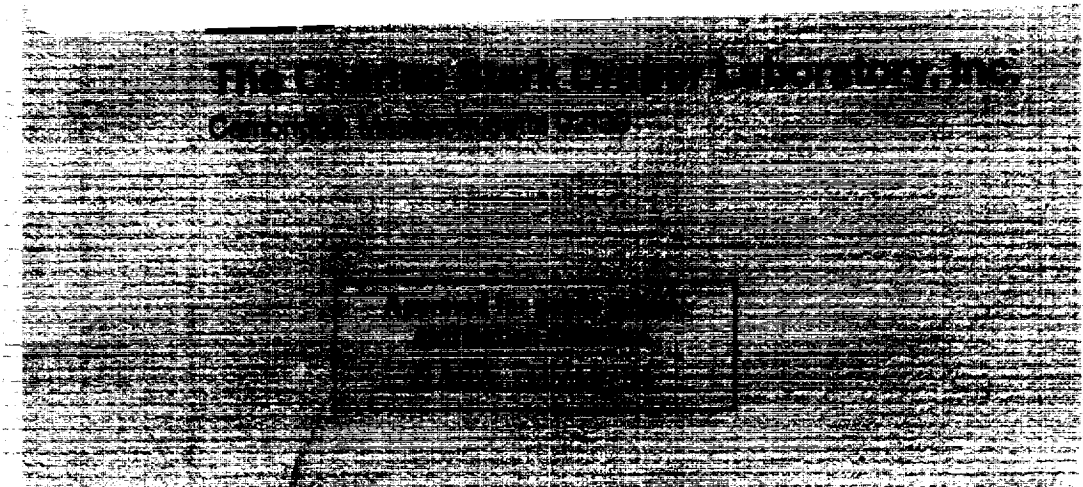
CSDL-C-5691

AIPS TECHNOLOGY SURVEY REPORT

February 1984

Prepared for the
Lyndon B. Johnson Space Center
of the
National Aeronautics and Space Administration
under
Contract NAS9-16023
Task Order 35



CSDL-C-5691

AIPS TECHNOLOGY SURVEY REPORT

February 1984

Approved: _____
Philip G. Felleman, Program Manager

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

ACKNOWLEDGEMENT

This report was prepared by The Charles Stark Draper Laboratory, Inc. under Contract NAS9-16023, Task Order 35, with the Lyndon B. Johnson Space Center of the National Aeronautics & Space Administration.

Technical portions of this report were written by the individuals named below, in the indicated subject areas.

- Architecture: Jaynarayan H. Lala
- Software: J. Barton DeWolf, Frederick C. Furtek, Paul A. Szulewski, and Mary E. Whalen
- Hardware: Alexander M. Kemp
- Reliability and Performance Methods: Paul R. Motyka
- Verification and Validation Methods: Robert E. Werner

Technical direction of the preparation of the report was provided by Robert N. O'Donnell, Technical Manager. The report was compiled and edited by Glenn Ogletree.

Philip G. Felleman, Program Manager

Publication of this report does not constitute approval by the NASA/JSC of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

ABSTRACT

This report documents the results of a technology survey conducted for the NASA/JSC by the CSDL during Phase I of the NASA Advanced Information Processing System (AIPS) program at the CSDL. The purpose of the survey was to ensure that all technology relevant to the configuration, design, development, verification, implementation and validation of an advanced information processing system, whether existing or under development and soon to be available, would be duly considered in the development of the AIPS.

The emphasis in the survey was on technology items which were clearly relevant to the AIPS. Requirements were developed which guided the planning of contacts with the outside sources to be surveyed, and established practical limits on the scope and content of the Technology Survey.

Subjects surveyed included architecture, software, hardware, methods for evaluation of reliability and performance, and methods for the verification of the AIPS design and the validation of the AIPS implementation. Survey requirements and survey results in each of these areas are presented, including analyses of the potential effects on the AIPS development process of using or not using the surveyed technology items.

Another output of the survey was the identification of technology areas of particular relevance to the AIPS and for which further development, in some cases by the CSDL and in some cases by the NASA, would be fruitful.

Appendices are provided in which are presented:

- Reports of some of the actual survey interactions with industrial and other outside information sources,
- The Literature List from the comprehensive literature survey which was conducted,
- Reduced-scale images of an excerpt ("Technology Survey" viewgraphs) from the set of viewgraphs used at the 14 April 1983 Preliminary Requirements Review by the CSDL for the NASA, and
- Reduced-scale images of the set of viewgraphs used in the AIPS Technology Survey Review presentation to the NASA monitors by the CSDL at the NASA Langley Research Center on 28 September 1983.

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

A technology survey has been accomplished in support of the NASA Advanced Information Processing System (AIPS) development to insure consideration of the broad spectrum of relevant technology. This report describes the survey as it was conducted, presents and discusses the information that was obtained as a result of the survey, and provides a preliminary assessment of the technology survey items relevant to the AIPS development. The assessment took into consideration the maturity of the technology, the risk associated with its use, and the potential benefit to be derived from its incorporation.

1.1 Purpose

The purpose of the technology survey was to insure that the latest technology available and to become available over the next few years was considered during the AIPS development. This was done to maximize the benefit of the AIPS development and to insure, to the extent feasible, compatibility between the AIPS architecture and future technology developments.

1.2 Scope

1.2.1 Document

This document is not intended to be a comprehensive description of the information obtained during the survey, but rather to provide a record of those items deemed to be of significance to the AIPS program. The Technology Survey description in Section 3, "Technology Survey" supplements and provides an interpretation of the presentation material used at the AIPS Technology Survey Review at NASA Langley Research Center (LaRC) on 28 September 1983. That presentation material (viewgraphs) is included for convenience in this report in Appendix D, "Technology Survey Presentation, 28 September 1983".

1.2.2 Survey

The survey covered, and was limited to, technology items that were known to be or were assumed to be relevant to the AIPS development program. These technology items were surveyed through contacts with industry, academia, the NASA, and the Department of Defense (DoD) as well as through extensive document searches. The contacts were made by way of

visits, telephone calls and letters as seemed appropriate to the situation.

Early in the program an initial set of technology survey items was developed and, following review, was amended to include those suggested by the AIPS Working and Peer Groups.

1.3 Technology Survey Report Description

Brief descriptions of the elements of this report are presented below.

1.3.1 Technology Requirements

The Technology Requirements section of this report (Section 2, "Technology Requirements") is concerned with defining the content and scope of the Technology Survey. While an initial set of survey topics and interview subjects were chosen these were both changed as the survey progressed, and as our knowledge of the survey topics and the needs of the AIPS program expanded.

1.3.2 Technology Survey

Section 3, "Technology Survey", presents and discusses the results of the survey which was conducted.

1.3.3 Technology Support

Technology items which are of particular relevance to AIPS and similar programs and which deserve accelerated development are identified in Section 4, "Technology Support." A proposed development agenda is set forth there.

1.3.4 Appendices

The appendices provide:

- Background information on certain interviews and other interactions (Appendix A, "Interaction Reports"),
- The results of the literature-search portion of the survey (Appendix B, "Literature List"),
- Pertinent excerpts from the Preliminary Requirements Review at Langley on 14 April 1983 (Appendix C, "Excerpts From 14 April 1983 Meeting at LaRC), and

- As noted previously, the presentation viewgraphs used at the AIPS Technology Survey Review at Langley on 28 September 1983 (Appendix D, "Technology Survey Presentation, 28 September 1983").

1.3.5 List of References

This listing is limited to those references referred to in the text of this report. Entries in the List of References may be also included in Appendix B, "Literature List."

SECTION 2

TECHNOLOGY REQUIREMENTS

2.1 Introduction

The AIPS is defined as a fault-tolerant, highly-reliable, flexible, distributed architecture suitable for a broad range of NASA applications. The AIPS will be developed using state-of-the-art technology while specifically allowing for the inclusion of emerging technologies. A survey of relevant technologies available for use in the AIPS development was planned as a Phase I activity. This section is intended to define the requirements that were established for the survey. For this purpose the discussion is divided into the five major topical areas of the survey:

(1) Architecture
(2) Software
(3) Hardware
(4) Reliability and Performance Methods
(5) Verification and Validation Methods

(In Section 3, "Technology Survey", the discussion and analysis of the results of the survey follows this same overall topical order, and are broken down further into subtopics and then further subdivided as necessary to address the details of the subjects surveyed and issues raised.)

2.2 Requirements

2.2.1 Architecture

The architecture issues relevant to the Advanced Information Processing System can be discussed in a framework defined by the following four subtopics:

(1) Processing Architecture - The processing architecture is concerned with the organization and control of the computational core of the system. The technology survey in this area was undertaken to examine the processing architecture of fault-tolerant computers of varying maturity. Operational avionics systems from commercial transport aircraft, military aircraft, the Space Transportation System (STS), and fault-tolerant systems in the industry were surveyed extensively. Experimental systems sponsored by the NASA and various branches of the Department of Defense were also targeted for survey. Finally, paper designs and other new approaches to

fault-tolerant and distributed computer architectures being developed at various academic centers were also closely examined.

Commercial fault-tolerant computers orinted towards transaction-type applications were excluded from this survey since their reliability goals did not match the AIPS requirements.

(2) Networking Architecture - The networking architecture is concerned with the interconnection of various computing elements in the system and their connection to sensors, effectors, displays, and other Input/Output (I/O) devices.

Technology survey targets in this area were the commercial Local-Area Networks (LANs), commercial aircraft buses, military avionics bus standards, and ongoing research projects at various NASA centers and in the industry. The aim here was to gather information on various networking topologies and operational alternatives.

(3) Fault Detection, Identification and Recovery: Computational Core - The AIPS is aimed at applications which require extremely high levels of reliability. The importance of detecting faults and recovering from them successfully cannot be overemphasized. A number of fault-tolerant systems - ground-based, airborne and spaceborne - were analyzed to collect and categorize the existing methods of detecting, identifying, and recovering from faults. Commercial main-frames, minicomputers, and laboratory computers were also examined from this viewpoint.

(4) Fault Detection, Identification and Recovery: Communication Media - Techniques and methods used to detect, isolate, and recover from faults in the communication media (between computers and between computers and Input/Output devices) are somewhat different from those employed in the computational core, that is, processors and memories. Also, one of the AIPS requirements is that various computing elements of the system be physically dispersable and that their interconnecting medium be secure and damage-tolerant.

Military and commercial avionics buses, Local-Area Networks, and NASA-sponsored laboratory research systems were some of the targets of the technology survey in this field of fault tolerance and damage tolerance.

2.2.2 Software

The second major topic for the AIPS technology survey is software. Software is expected to be a major component of the AIPS system, and software technology has been going through a period of rapid change. The following subtopics were selected for emphasis:

(1) High-Order Languages (HOLs)
(2) Fault-Tolerant Software
(3) Distributed Operating Systems
(4) Software Tools and Methods

These items were selected based on anticipated technology needs for the AIPS system. It is desired to use a high-order language for AIPS, and to make a reasoned selection from among several possibilities. The program's emphasis on reliability makes the issue of software reliability in general and fault-tolerant software in particular extremely germane. The design of the AIPS operating system is expected to present many challenges, particularly in light of the AIPS requirements for accomodating future growth. Finally, it is desired to be able to make use of the most appropriate and useful tools and methods in developing software for the AIPS system.

The following sections describe the survey requirements that were initially identified for each of the foregoing subtopics.

(1) High-Order Languages - Given the strong desire to use a high-order Language for AIPS, the technology survey tried to supply a rationale for selecting one. It was agreed to use a standard language. Emphasis was placed on the DoD standard, Ada[1]. Hal is a NASA standard but is currently unsupported except in a maintenance mode by the Shuttle program.

The critera that were selected to evaluate the languages and their possible use in AIPS were:

Their availability,

The availability of related tools to support development and testing,

The impact of their use on reliability and growth, and

Their run-time efficiency.

(2) Fault-Tolerant Software - Software fault tolerance, unlike hardware fault tolerance, cannot be achieved by simple replication of identical elements. Because software errors stem from faulty designs, fault tolerance can be achieved only through redundancy of design. This means that multiple copies of a program are independently developed and coded, with all copies starting from a specification of the program's requirements. The specification should state the requirements completely and unambiguously while at the same time constraining the implementation as little as possible. Where possible, different algorithms and different programming languages (or translators) should be used for each version of a program.

Three approaches to achieving software fault tolerance were identified: N-version programming, recovery blocks, and backup software. It was desired to evaluate them according to the following criteria:

---

1 Ada is a registered trademark of the U.S. Government (Ada Joint Program Office). See page A-1, Appendix A, for additional information.

Effectiveness in the intended environment,
Prior experience,
Development cost, and
Operational cost.

(3) Distributed Operating Systems - The AIPS consists of a dispersed collection of interacting processors. The AIPS operating system must therefore oversee not only the operation of each individual processor but also the collective operation of the entire system. The role of the operating system is as both servant and supervisor. As a servant, the operating system provides to the application software such facilities as interprocess communication and data-base management. As a supervisor, the operating system makes decisions regarding such activities as task scheduling and reconfiguration. Implementation of these functions is a significant challenge, and the demands are made even more severe by the twin requirements of fault tolerance and real-time operation. Although there does not exist an operating system meeting all of the AIPS needs, a number of existing systems have been examined with an eye towards incorporating certain elements into AIPS.

(4) Software Tools and Methods - Software tools are powerful productivity and quality aids used by software developers and managers. Specifically, these tools are computer programs which aid in the specification, construction, testing, analysis, management, documentation, and maintenance of other computer programs. Thus, tools include traditional tools of the programmer (e.g., compilers, editors), more recently developed tools (e.g., requirements and design aids, testing, configuration-management tools, and development environments), and tools currently in the research stage (e.g., automated quality analyzers, and formal specification and verification tools). Methods support the application and use of these tools during the software development life cycle. Development and maintenance of any large software system requires mature tools and methods to aid development personnel and managers.

Since the AIPS is envisioned as a large real-time and distributed processing system, it has the potential to have a complex software architecture. Modern software development practice has provided ample evidence that good tools and methods increase software reliability, good communication, and developer productivity, and at the same time reduce development and maintenance costs, and enhance management visibility into the evolving software product. Thus, the use of the most modern and effective tools is essential to the AIPS development.

In order to determine specific tool and method requirements for AIPS, developers of both real-time software and network software were interviewed and a literature search in the software tool category was undertaken.

2.2.3 Hardware

The third major topic for the AIPS Technology Survey is hardware. The following subtopics were selected:

(1) Processor Instruction-Set Architecture
(2) Semiconductor Technology
(3) Power Distribution and Control
(4) Mass Memories

These subtopics are discussed separately below.

(1) Processor Instruction-Set Architecture (ISA) - This survey covered implementations of processor instruction-set architectures and how they relate to the AIPS requirements. Included in the survey were:

The currently available commercial microprocessors,

Commercial microprocessors announced but not yet in production,

Future trends in commercial microprocessors, and

Present state of implementations of military specifications.

In surveying hardware implementations of an instruction-set architecture it was necessary to apply an evaluation criteria to each such implementation to determine its suitability to the AIPS.

When reviewing a commercial microprocessor it was necessary to evaluate its suitability to the different AIPS requirements. Of prime importance was its industrial acceptance which would assure its longevity and its being the baseline for future compatible developments. Other important considerations included the maturity of the technology and the ability to meet environmental requirements. In systems such as the AIPS a prime requirement is the availability of high-order languages.

The AIPS technology survey had to be cognizant of developments in instruction-set architectures which are defined by military specifications. All the considerations deemed important for commercial instruction-set architectures were required to be applied to the military specifications also. In addition, the AIPS program had to assess the viability of the specifications as well as examine the different physical implementations of each.

(2) Semiconductor Technology - In choosing a technology for an AIPS implementation it is necessary to look at the present and predicted future status of the semiconductor manufacturing processes and how they interface with the project needs. All semiconductor families had to be evaluated and appropriate tradeoffs made for a particular mission. The environmental characteristics of the semiconductor technology had to be matched to the selected AIPS environment. Questions to be answered for each included:

Is the technology mature?,
Does its speed match the specific application?,
Is the power compatible with the available power source?,
Is the quality acceptable?,
Is the technology available from multiple viable sources?;
Are components available on the qualified parts list?, and
What is the estimated future of the technology?

In examining the semiconductor field an important task was to determine if manufacturers are migrating towards any particular technology for the future.

(3) Power Distribution and Control - The AIPS requires a highly reliable and fault-tolerant power generation and distribution system. Areas that are important to the AIPS are:

Multiple power sources,
Multiple power buses,
Power grids/networks,
Power system control,
Power conditioning, and
Fault-containment schemes.

The AIPS technology survey reviewed existing literature for contributions to reliable power generation and distribution. Interviews were carried out with people responsible for the designs of power systems for spacecraft and aircraft.

(4) Mass Memories - The AIPS requires mass memory for the storage of program and data. The memory field is changing dynamically. It was necessary to examine the current availability as well as the predictions and trends for the future. The technology survey for the AIPS examined semiconductor, bubble, magnetic tape, and magnetic and optical disc storage devices. Each had to be evaluated for maturity, speed, capacity, physical volume and environmental characteristics.

In the survey, manufacturers were contacted and the literature examined to determine the present and projected states of memory devices.

2.2.4 Reliability and Performance Methods

The fourth major topic of the AIPS Technology Survey is Reliability and Performance Methods. This survey was undertaken to ensure that the latest evaluation methodologies available are made use of during the development of the AIPS system. It is also desired to minimize methodology development costs for the AIPS program. The survey was conducted by contacting relevant parties in industry, academia and the government. A thorough appraisal of the technical literature was also part of the survey.

The scope of the methodologies survey included the following technical subtopics:

(1) Hardware Reliability, Maintainability, Availability (RMA)
(2) Software Reliability
(3) Computer System Performance
(4) Performability - A Combined Consideration of Performance and Reliability

The major goal of this effort was to assess the applicability and maturity of the methodologies within each of these areas with regard to the evaluation of AIPS architectures. Three subgoals were defined within this overall goal. They were:

To compare generic approaches for evaluating AIPS-type systems within each of the defined technical areas,

To define measures of merit for quantitatively assessing the AIPS system, and

To compare existing evaluation mechanizations for their applicability to AIPS.

2.2.5 Verification and Validation Methods

The fifth major topic of the AIPS Technology Survey pertained to the available methods for the verification and validation (V&V) of developed articles and products with respect to their satisfaction of the requirements and constraints imposed on them. For the sake of communication, the following definitions are offered:

Verification: The process of establishing that the developed article (software, hardware, system) satisfies both the system requirements and its individual requirements at each step of the development.

Validation: The process of establishing that the developed product satisfies the system requirements and complies with the vehicle requirements. The objectives are to demonstrate this compliance under operating conditions and the absense of undesired effects. The process may include simulation and/or flight test.

With these definitions and the evolving AIPS requirements in mind, a survey of various programs was instituted to determine the state of the art of verification and validation as applied to current systems. This was performed through the media of telephone conversations, visits to various facilities, literature search, personal contacts with in-house projects, and personal experience. The emphases in this survey task were to compile the survey results, to identify issues raised by an examination of the results, and to distill from the information acquired and reduced any new (learned) information of use in developing the AIPS V&V methodology.

SECTION 3

TECHNOLOGY SURVEY

3.1 Introduction

The technology survey, as previously stated, was performed by contacts with industry, academia, the NASA, and the Department of Defense, and through a comprehensive literature search. The baseline technology requirements were modified as the survey and AIPS architectural concepts matured. This provided a more fruitful interaction with the technological community than might otherwise have occurred with a fixed agenda.

The discussions that follow are intended to supplement and interpret the presentation material found in Appendix D, "Technology Survey Presentation, 28 September 1983". The reader is encouraged to refer regularly to the presentation material in Appendix D as the related text is perused here, to obtain a more comprehensive understanding of the survey.

3.2 Survey Areas

The discussion of survey results in this section follows the same breakdown into five major topical areas as was done in Section 2, "Technology Requirements". The five topics are:

(1) Architecture (see "Architecture," below)
(2) Software (see "Software" on page 3-13)
(3) Hardware (see "Hardware" on page 3-17)
(4) Reliability and Performance Methods (see "Reliability and Performance Methods" on page 3-20)
(5) Verification and Validation Methods (see "Verification and Validation Methods" on page 3-28)

3.3 Survey Results

3.3.1 Architecture

The results of the architecture technology survey can be discussed in a framework defined by four subtopics:

(1) Processing Architecture
(2) Networking Architecture

(3) Fault Detection, Identification and Recovery (FDIR): Computational Core
(4) Fault Detection, Identification and Recovery: Communication Media

These are discussed and conclusions drawn in sequence below.

3.3.1.1 Processing Architecture

The processing architecture can be discussed at the system level, processor level, and Instruction-Set level:

(1) System-Level Architecture - The system-level architecture is concerned with the overall organization and control of all the computing elements in the system. The systems surveyed can be categorized at this level as centrally oriented, distributed or federated. The centrally oriented systems have centralized program, data base, and control. Examples of this type of system are the CSDL/Collins Fault-Tolerant Multiprocessor (FTMP), SRI/Bendix Software-Implemented Fault Tolerance (SIFT), and others shown in the architecture viewgraphs in Appendix D. The FTMP and SIFT computers were sponsored by the NASA Langley Research Center. They are now operational in the Avionics Integration Research Laboratory (AIRLAB) at that NASA center. The object of this program was to demonstrate a technology necessary to safely fly a commercial transport aircraft which is statically unstable and requires a full-time, full-authority digital flight control system.

These systems and others mentioned in Appendix D cannot really be classified as truly centralized systems. A totally centralized system would also have only one computing element. Therefore, the examples cited here can only be classified as centrally oriented rather than centralized systems.

The next category of system-level architecture is the distributed systems. A totally distributed system as defined by Enslow [1][2] would have to satisfy several requirements. A totally distributed system must have a multiplicity of resources which are freely assignable to functions on a short-term basis. The multiplicity of resources should be transparent to the user. The active system components should be interconnected via a network that utilizes cooperative protocols. The system control is through multiple cooperating autonomous Operating Systems (OSs). Finally, the data base is completely replicated in all subsystems or it may be partitioned and may or may not have a central master copy or master directory. Although none of the systems surveyed fit this category in the sense of satisfying all of these requirements, there were some systems which had a flavor of these charateristics. One of these is the Distributed System Breadboard (DSB) under development at the NASA Johnson Space Center. It consists of a number of computers interconnected by a ring network. The goals for the Operat-

[2] Numbers in brackets refer to similarly numbered entries in the List of References found at the end of this report.

ing System for the DSB are consistent with the narrow definition of the distributed system of Enslow.

The viewgraphs in Appendix D show a number of systems under each of the previously defined two categories, viz., centrally oriented and distributed. Actually, all the systems surveyed are hybrids in nature. That is, they have some attributes from each of the categories. One cannot therefore classify them as purely centralized or purely distributed, etc. But they may be closer to one than the other.

The last category is the federated systems. Examples of these are the Boeing 757/767 avionics and the STS avionics. The federated systems differ from the distributed in several areas. One is the assignment of resources to functions. In a federated system resources are dedicated to functions. The second is the system control. The federated system control is vested in subsystems through independent operating systems. Finally, the data base in a federated system is partitioned with no central master copy or directory.

The two examples cited above fit in this category if one includes all of the vehicle avionics such as flight control computers, navigation and guidance computers, engine controllers and so on.

Although one may disagree with the definition and categorization of the systems, that is not an important issue. The survey was undertaken with the aim of learning about the charateristics of systems that have been built, designed or proposed rather than to place them in some predefined categories.

A major conclusion that can be drawn from this part of the architecture technology survey is that at the system level there is a diversity of approaches to designing an architecture. However, there is no operational system consisting of multiple computers where functions migrate among computers at will and the user is unaware of the multiplicity of resources. The only example of a proven architecture that comes close to these goals is the multiprocessor. Although a multiprocessor is strictly speaking not a multi-computer it does have a multiplicity of processing resources which are freely assignable to functions on a short-term basis, and these actions are transparent to the user.

(2) Processor-Level Architecture - The processor-level architecture is concerned with the interconnection of processor, memory, and I/O devices. For redundant computers it also describes the interconnection and communication between redundant elements of a computer. At the processor level one can classify the systems surveyed either as uniprocessors or multiprocessors. The results of the technology survey in these two areas are described separately below:

- Uniprocessor Architecture: There are a number of examples of uniprocessors, ranging in maturity from prototypical to opera-

tional. The SRI SIFT computer can be configured to operate as a uniprocessor. Its redundant processing elements are fully cross-strapped and are frame synchronized. All the hardware faults are handled in software. Simplex source congruency is also implemented in software. The CSDL Fault-Tolerant Processor (FTP) is another example of a prototypical uniprocessor. It also consists of fully cross-strapped redundant processors. The redundant channels are tightly synchronized. The hardware faults are detected using hardware. The fault isolation and recovery functions are implemented in software. Simplex source congruency is mostly handled by hardware.

Major examples of experimental fault-tolerant uniprocessors are the F-8 Digital Fly-by-Wire System (DFWS), the A-7 DIGITAC I and II, and the F-16 Advanced Fighter Technology Integration (AFTI) Flight Control System.

The F-8 Digital Flight Control System was sponsored by the NASA Dryden Research Center and designed and built by the CSDL using three AP-101 computers. This was the first digital system to fly in an aircraft without a mechanical backup. The only backup system is a triplex analog computer. This system is also based on a frame-synchronous architecture.

The A-7 DIGITAC program is sponsored by the Flight Dynamics Laboratory (FDL) of the Air Force Wright Aeronautical Laboratories (AFWAL). Phase I of this program consisted of a dual-monitored digital flight control system designed to augment stability and perform pilot-relief autopilot functions. The mechanical linkages were not removed and the stability augmentation function was not flight-critical. In Phase II the same dual configuration has been retained but uses a multiplex bus inside the basic control loop, which is the first time that has been done.

The AFTI/F-16 has been developed for the Flight Dynamics Laboratory by General Dynamics to demonstrate advanced technology concepts. The flight control system is a fully cross-strapped triplex digital system. The three channels were initially configured to operate totally asynchronously. However, during flight tests enough cross-channel data exchange has been added to make the system virtually frame synchronous.

Among operational fault-tolerant uniprocessors, the most outstanding example is the STS Digital Processing System (DPS). This is the first flight-critical system to fly with no mechanical or analog backup of any kind. The only backup it has is another digital computer with a different and simplified version of the primary software. The primary DPS consists of four AP-101 computers which are organized as a quad-redundant computer during ascent and reentry phases of the mission. Software is used to synchronize frames of the four computers. During on-orbit operation, the four computers are configured in a multi-computer fashion. In one of these configurations,

one pair is assigned critical tasks while the remaining two are assigned to different, non-critical tasks.

Examples of other operational uniprocessors include the Digital Flight Guidance System on the McDonnell-Douglas MD-80 commercial transport aircraft. This is Sperry's dual-redundant fail-safe system with the two channels operating asynchronously. The Boeing/Collins Flight Control System on the 757/767 series of transport aircraft is a triplex, frame-synchronous fail-operational system. The General Electric Flight Control System on the Navy F/A-18 aircraft is a quad-redundant asynchronous system that is almost totally flight-critical. The aircraft does have a mechanical backup but can be operated manually only in a very limited flight regime.

The CSDL designed Fault-Tolerant Processor is now in production by Woodward Governor Co. for use as a ground-based turbine control system for applications requiring low maintenance and high reliability.

The Electronic Switching System No.1A designed and operated by the American Telephone and Telegraph Company (AT&T) is the largest and most recent in a series of digital switching systems developed over the last 30 years. It is designed to handle up to 100,000 calls per hour. It is a dual-redundant, tightly synchronized computer that is supposed to have an extremely high availability but not necessarily very high reliability. The availability goal for this system which is now widely in use throughout the country is a down-time of no more than 3 minutes a year or a total of no more than 2 hours in 40 years. The reliability goal, on the other hand, is to have no more than 0.02% of the calls be handled incorrectly. That is, a failure rate of 20 incorrectly handled calls per hour is acceptable.

The conclusion from this part of the technology survey is that Redundant Uniprocessor Architecture is a maturing technology. The most common approach to handling the redundancy is to have a level of synchronization between parallel channels. The totally asynchronous systems are in a minority and not being pursued in newer designs.

- Multiprocessor Architecture: Some examples of the multiprocessor systems have already been cited in the system-level architecture technology survey. These include the FTMP and the SIFT computers. The FTMP relies on tight synchronization and exact bit-for-bit comparison and voting of redundant processors and memories for fault detection and masking. Dynamic reconfiguration is used to replace failed modules with spares or gracefully degrade the system if no spares are available. The SIFT approach also relies on exact bit-for-bit comparison and voting to achieve high coverage. However, comparison and voting logic is implemented in software. The overheads of doing this in software, especially when simplex source congru-

ency overheads are also included, makes this an impractical or a very expensive approach.

Other examples of the multiprocessor architecture for real-time systems include the crossbar-type system at the Ballistic Missile Defense Advanced Technology Center (BMDATC) in Huntsville, Alabama. The present configuration consists of 6 processors (Z8001) connected to a bank of 16 memory modules through a crossbar. This system is not directly relevant to AIPS since it has very little fault tolerance designed into the system for protection against processor, memory or crossbar failures. The BMDATC application, of course, has very different requirements in this area compared to the AIPS requirements. Their system has to run correctly for only about 10 minutes to 1 hour although it must be able to cold start correctly and with sufficient operating components to do the job after being dormant for years.

Another example of the crossbar-type multiprocessor is the C.mmp, a research system at Carnegie Mellon University (CMU). It consists of 16 slightly modified PDP-11's connected to 16 memory modules. Unlike the BMDATC crossbar where processors can talk to each other only through the shared memory, in the C.mmp a time-shared multiplexed bus (interbus) with controllers on each of the individual processor unibuses connects processors. But this system also has very little high-coverage fault tolerance designed into it.

Other multiprocessors operational at CMU include the C.vmp and the Cm*. The C.vmp is a "Voted Multi-Processor". It consists of three processor-memory pairs (LSI-11s), each pair connected by a bus. It can operate as a multiprocessor with each of the three computers working on a separate program. Under control of an external event or one of the three processors it reconfigures itself into a voted uniprocessor.

The Cm* is another multiprocessor system operational at CMU. The basic building block of the Cm* architecture is the computer module, Cm. It consists of a processor (LSI-11) and 4k to 124k of primary memory. Several Cm computers are connected together via a local bus and form a cluster. The Cm* consists of several clusters connected via intercluster buses.

Finally, the Pluribus multiprocessor designed in 1972 by Bolt Beranek and Newman Inc. is a reliable, high-capacity Interface Message Processor (IMP) used by ARPANET at each of its switching nodes. It consists of seven dual processors (Lockheed SUE minicomputers) with each pair sharing 2 modules of 4K words of local memories on a single bus structure. Through a bus arbiter and extender, these dual processors can access 8K words of shared memory. The main goal for Pluribus was to maximize availability rather than aim for error-free operation. For example, if a message is lost, its recovery is not attempted by Pluribus but left to the network.

(3) Instruction-Set Architecture (ISA) - At the lowest level, the processing architecture consists of the Instruction-Set Architecture. The ISA defines the interface between the hardware and the software.

A taxonomy based on Flynn's work [2] is shown in the Viewgraphs of Appendix D. The parts relevant to the AIPS program are the commercial and MIL-STD Von Neuman-type Instruction-Set Architectures.

The Air Force standard is the MIL-STD-1750A. It has several limitations from the viewpoint of AIPS requirements. It has a limited address space. Only 64K words can be directly addressed by a process although the total processor address space can be extended to a megaword using the Memory Management Unit. It does not support virtual memory. Several manufacturers are implementing this instruction set on a single chip although none is currently available.

The Army counterpart of this is the MIL-STD-1862 or Nebula architecture. It has a 32-bit address space and is a more modern and up-to-date architecture. But it is not likely to be implemented in silicon for 3 to 4 years.

There are a number of commercial microprocessors which are attractive alternatives. Among these is the Motorola M68010 which is a 16-bit architecture but has 32-bit address space and supports virtual memory. It is available and it is likely to be one of the three de-facto standards of the industry. Another attractive feature of this microprocessor is the wide availability of support tools, software, floating-point co-processor and so on. Currently Ada compilers are available for its predecessor, the M68000, and Ada compilers for the M68010 are planned for the near future. It is also upward-compatible with the M68020, the planned 32-bit version of this processor.

Processing Architecture Conclusions - While none of the systems surveyed satisfies system-level AIPS requirements, elements of them will be considered for inclusion in AIPS.

At the processor level there are several promising candidates. If the AIPS requires a fault-tolerant uniprocessor, the architecture chosen would use an instruction-synchronous approach with fault detection, masking, and source congruency implemented in hardware. If a multiprocessor is to be used as a component of the AIPS then the candidate architecture is an instruction-synchronous homogeneous multiprocessor.

At the ISA level there are several options; the MIL-STD-1750A or the commercial state-of-the-art microprocessors such as M68010, etc.

3.3.1.2 Networking Architecture

The Networking architecture is concerned with the interconnection of computers and their connection to the sensors, effectors, displays, and

other I/O devices. There are four aspects of this architecture:

(1) Topology
(2) Network Operational Alternatives
(3) ISO Open Systems Interconnect Model
(4) Network Software Interface

These aspects are discussed and conclusions drawn in sequence below:

(1) Topology - The interconnection scheme can be described topologically as a bus, ring, mesh, tree or full cross-strapping. Examples were found in each of these categories in the systems that were surveyed.

The most common form of interconnection scheme is a linear bus. The distinction between most buses is their protocol. For example, commercial avionics use broadcast buses such as ARINC 429. A broadcast bus consists of one transmitting unit and a number of receiving units. The military avionics use a command/response bus such as the MIL-STD-1553B. A unit transmits only when it is commanded to do so by the central bus controller. Finally, most Local-Area Networks (LANs), notwithstanding the name "Network", really consist of a number of terminals connected by a linear bus. They typically use a contention bus. Another example of a contention bus is the FTMP system bus that interconnects all processors and memories in the computer.

The ring topology is also used by some LANs such as the Prime Net, Corvus Omninet, and Apollo Domain. The ring topology makes it easier to implement token passing as the bus arbitration scheme. Other examples of this topology are the Johnson Space Center's distributed system testbed which connects a number of computers in a ring. Presently copper wires are used for interconnection but these are slated to be upgraded to high-performance optical fibers. The CSDL-developed and -designed Industrial Machine Controller (IMC3) for FIAT consisted of a 5-computer ring that was delivered as a prototypical system to FIAT in 1976.

A more complex topology than a ring is a mesh which provides several alternative paths between any two points in the system. The CSDL mesh concept which uses circuit-switched nodes was developed as a fault- and damage-tolerant network to provide a secure means of communicating to sensors and actuators in an aircraft. A prototype of the Dispersed Sensor Processing Mesh is now operational in the F-8 Ironbird at the NASA Dryden Research Center. Another example of the mesh network is the ARPANET which connects a number of mainframe computers around the country through message-switching nodes.

The most secure way to interconnect computers is to fully cross-strap them as was done in the SRI/SIFT computer. In the SIFT computer point-to-point dedicated links were provided between each pair of processors.

(2) Network Operational Alternatives - If a mesh network or fully cross-strapped topology is selected then such a network can be operated in two different ways. Packet switching can be used to transmit messages from one point to the next as is done in ARPANET. Or circuit switching can be used as is done in CSDL mesh, BMDATC crossbar multiprocessor, and Bell switching centers. The packet-switching method has a built-in delay due to store and forward strategy while a circuit-switching mechanism has very little delay at the switching centers due to a virtual bus-type path between two points.

Another dimension of operational alternatives is the multiplexed use of the communication medium. Time-Division Multiple Access (TDMA) is the most common way of letting a number of users share the same bus. The protocol whereby contention between multiple users is resolved is what distinguishes various TDMA schemes. In command/response- type multiplexed buses such as MIL-STD-1553, a central bus controller resolves contention. LANs such as Ethernet, Wangnet, and Z-Net use the Carrier-Sense Multiple Access scheme with Collision Detection (CSMA/CD). A user desiring to transmit waits until the carrier-sense circuitry detects a quiescent state on the bus. A transmission is initiated at that point. If several users happen to transmit at the same time then collisions on the bus are detected and everyone stops transmitting. After a random wait new transmission is begun.

Hyperchannel and Omninet use a variation of this scheme, called the CSMA/CA (...with Collision Avoidance) which avoids collisions.

Other bus-arbitration schemes include token passing which requires at least a ring topology to be efficiently implemented, and the Laning Poll.

Another way to multiplex use of a single communication link among several users is to provide each user with a different frequency. Cable TV is an example of this technology, called Wavelength- Division Multiple Access (WDMA). Up to 45 different frequencies are transmitted simultaneously on the CABLE TV Networks.

The NASA Langley Research Center is sponsoring research in the area of fiber-optic WDMA. They have successfully multiplexed, transmitted, and demultiplexed four frequencies on a single optical fiber. Their goal is to increase this to 32.

(3) ISO Open Systems Interconnect Model - The seven layers of the Open Systems Interconnect Model of the International Standards Organization are shown in the viewgraphs of Appendix D. The aim of this standard is to make changes in the lower layers transparent to the part of the system concerned with higher layers.

The two lowest layers, the Physical and the Data Link, define the interface with the physical channel and are hardware-intensive. The next five layers are software intensive. At the present

time none of the systems in existence has implemented all seven layers of the protocol. The Ethernet LAN now has four of these layers implemented. It is not clear at this point in time how much overhead will be required to support this standard.

(4) Network Software Interface - The last topic of the technology survey under Networking Architecture was the software interface of the network. The network operating system can be a single distributed operating system or a set of multiple communicating operating systems.

Networking Architecture Conclusions - Two conclusions can be derived as a result of the technology survey conducted in this area:

First, as far as the interconnection topology is concerned, the multiplex bus is vulnerable to faults and damage and the full cross-strapping is the most secure interconnection although it is also the most complex.

Second, as far as operational alternatives are concerned, token passing and Laning Poll are good candidates for bus arbitration since these are deterministic as opposed to CSMA/CD which is not. Finally, circuit switching is preferable to packet switching due to very little latency of the former scheme.

3.3.1.3 Fault Detection, Identification and Recovery: Computational Core

There is a diversity of methods and techniques used to detect, identify, and recover from faults in the processor, memory part of digital systems. These can be classified in four major categories as follows:

(1) Temporal Checks
(2) Integral Checks
(3) Diagnostic Checks
(4) Replication, Comparison and Voting

These categories are described in sequence below. The architecture viewgraphs found in Appendix D give examples of systems that use these FDIR methods.

(1) Temporal Checks - The temporal checks consist of time-related checks. Watchdog timers are used extensively in computers to detect processor failures. Typically, if a certain address is not written-to every few milliseconds an interrupt is generated which can be used to reset the processor or fail-safe the processor. Watchdog timers can also trap runaway software faults.

Most central processing units (CPUs) use bus timeouts to trap faults in which a memory does not respond.

Instruction re-try is used as a way of recovering from these faults by mainframe computers.

(2) Integral Checks - Integral checks are checks that are built into the basic number representation. Codes are an example of this type of fault-detection method. The simplest and most common representation of coded information is the parity bit. It can detect single-bit failures. Parity is widely used in memory of most minicomputers and in the data path of most mainframes. Hamming code is an example of a more complex form of code. It can not only detect but also correct errors. Most mainframe computers use Hamming code or some other variation of it to detect and correct memory faults.

Arithmetic codes (i.e., codes that are preserved through arithmetic operations on variables) were used some years ago when hardware was still quite expensive and it was not possible to replicate the whole computer.

Morphic logic or self-correcting logic is another form of integral check and has been used in the Self-Correcting Computer Modules (SCCM) designed by the NASA Jet Propulsion Laboratory (JPL).

(3) Diagnostic Checks - Diagnostic checks are undertaken to uncover already existing faults or latent faults. Read-Only Memory checksums, CPU opcode tests, tests of voters etc. are examples of diagnostic programs used to uncover faults in parts of the computational core which is not exercised by regular programs.

(4) Replication, Comparison and Voting - The previous three classes of fault detection and correction methods have a very low coverage, ususally less than about 90 percent. To achieve the very high coverage required for life-critical systems the most common method is to replicate the hardware. Hardware can be replicated at a low level or at a high level. An example of the former is the Raytheon/Air Force Spaceborne Fault-Tolerant Computer (SFTC). This computer utilized redundant elements of the CPU rather than the whole CPU replication to achieve high fault-detection coverage. Since off-the-shelf components are ruled out by this approach, a set 22 custom LSI chips was necessary to implement this architecture.

Most other fault-tolerant computers use high-level replication which means replication at the CPU, Memory level or even at the computer level. What distinguishes these architectures from each other is the way replicated channels are operated. Systems such as the MD-80 Digital Flight Guidance System uses two identical computers but they operate such that their results are only approximately equal even under no-fault conditions. Other examples of the approximate replication are the AFTI/F-16 Flight Control System and the NASA Ames Redundant Asynchronous Multi-Processor System (RAMP).

Systems such as the CSDL FTMP, the SRI SIFT, the space shuttle DPS, and the CSDL FTP are examples of exact or congruent replication. That is, the results of the redundant processors or computers match exactly or bit-for-bit when there are no faults in the

system. The exact replication approach is the most commonly used approach due to its higher coverage.

Another way to operate redundant channels is to place them in the standby mode. The Bell ESS No. 1A is a dual-redundant system but at any given time only one computer is in charge. Transition to the backup computer is made when the primary computer fails. The Voyager spacecraft's Command Computer Subsystem (CCS) is also an example of this approach.

Conclusions pertaining to the FDIR/Computational Core survey are combined with those pertaining to the FDIR/Communication Media survey which follows, to form overall FDIR conclusions.

3.3.1.4 Fault Detection, Identification and Recovery: Communication Media

The methods used to detect and recover from faults in the communication media can be classified in the three categories discussed below:

(1) Temporal Redundancy - This is a form of redundancy used in time. Two most notable examples of this are the CRC checks used in Local-Area Networks and the Bose-Chaudhri (BC) codes used in communication networks. The CRC check consists of a form of check-sum transmitted at the end of each message. The BC codes are used to correct for burst errors or multiple errors which are quite prevalent in communication networks.

(2) Spectral Redundancy - Spectral redundancy implies sending the same information over the same communication medium but using different frequencies. Wavelength-Division Multiplexing, if used to transmit the same data on different frequencies would be an example of this. Although this has been proposed it is presently not used anywhere due to lack of maturity of Wavelength-Division Multiple Access (WDMA) technology and also due to the highly correlated failure of all the redundant information.

(3) Spatial Redundancy - Spatial redundancy simply means redundant communication links. Typically most avionic systems use redundant buses. Another form of spatial redundancy is the mesh network such as the ARPANET, the AT&T telephone network and the CSDL mesh network.

Conclusions are combined with those for FDIR/Computational Core in the following subsection.

3.3.1.5 Fault Detection, Identification and Recovery: Conclusions

The FDIR conclusions, covering both computational core and communication media, are:

Replication is the most effective means of providing very high coverage.

Integral checks are suitable for large memories.

Spatial redundancy is most suitable for communication media.

Temporal and diagnostic checks are helpful in uncovering latent faults and enhancing fault detection coverage.

Codes are helpful in enhancing communication fault detection coverage.

3.3.1.6 Architecture Technology Summary

A very diverse technology base from various NASA centers, Department of Defense branches, Industry, and Academia has been examined. Operational systems, prototypical systems, and paper designs yet to be implemented in hardware were all examined in great detail. It was anticipated that this comprehensive survey of the fault-tolerant architectures would form the basis for the design of the AIPS.

3.3.2 Software

The software technology survey was carried out in four subtopical areas, namely:

(1) High-Order Languages
(2) Fault-Tolerant Software
(3) Distributed Operating Systems
(4) Software Tools and Methods

These subtopics are discussed in sequence below.

3.3.2.1 High-Order Languages

During the survey of compilers, emphasis was placed on the Ada language.

Ada is a high-order language (HOL) designed and developed for the Department of Defense (DoD) for use in embedded systems. The DoD has designated Ada as the DoD's standard language for embedded systems and is dedicating a large quantity of resources to its development. In an attempt to reduce the costs associated with the development, testing, and maintenance of the software for embedded systems, the DoD has committed an enormous effort to the design of the Ada language specification and the Ada Program Support Environment (APSE).

A requirement of the AIPS program is that the software for the program be done in a HOL. The general nature of the AIPS program, the fact that we are looking to future technologies, and the possible application to DoD programs, makes Ada a candidate to be that HOL.

We considered several other languages, their suitability and their availability during the survey. Those languages are:

Jovial,
Hal,
Fortran,

C, and
Pascal.

We have a good deal of in-house experience with all of these languages. Compilers are available for all of them, with code generators for various of the targeted processors that have been considered for AIPS. The Hal language, which is a NASA standard, is unsupported except for maintenance in the Shuttle program.

It is too soon to decide whether to select Ada as the implementation language for the AIPS program. Ada schedules have been slipping during the period of this survey, but even if delivered on time, Ada is a new language with many new and untested concepts. However, it does seem reasonable to use an Ada program-design language (PDL) on AIPS, and to take care to design the AIPS software in such a way that a conversion to Ada in the future is not impossible.

Visits and telecons were made to various Ada builders and vendors. We talked to and listened to demonstrations by Ada educators. We tried to find users of the various compilers and APSE's. This last was difficult given the immaturity of Ada.

3.3.2.2 Fault-Tolerant Software

Three approaches to fault-tolerant software have been identified:

(1) Recovery Blocks
(2) N-Version Programming
(3) Backup Software

Recovery blocks were pioneered by Brian Randell and his associates at the University of Newcastle upon Tyne, while Algirdas Avizienis has been the prime mover behind N-Version programming, first at JPL and now at UCLA. Backup software is a common approach to software fault tolerance in a variety of flight/space systems including some designed at CSDL.

In the **recovery-block** approach, there exists an ordered list of alternatives for a program, the first alternative being called the primary alternative. The alternatives are executed in order as required. Upon termination of an alternative, an acceptance test is performed to determine the "appropriateness" of the output. If the test is successful, then the recovery block terminates with the output. If unsuccessful, then the next alternative in line is placed into execution after the local state has been reset. If all alternatives have been attempted without success, then the recovery block terminates with an error code.

N-version programming, like recovery blocks, has multiple copies, or versions, of a program. But unlike recovery blocks, there is no priority assigned to the different versions and all versions are executed upon a call to the program. (Execution may be sequential or parallel depending upon the computational resources available.) The final output of the program is determined by "vote" of the different versions. N-version programming can be integrated in a natural way with N-version hardware to provide for both hardware and software fault tolerance.

In the case of **backup software**, a separate backup system is maintained in the event that an emergency situation makes execution of the primary software untenable. Once control is passed to the backup software, the backup mode remains in effect until remedial action can be taken to correct the primary software. Typically, the backup system provides only those functions essential to safe operation.

The three approaches have various limitations and shortcomings and these are summarized in the software viewgraphs in Appendix D. In terms of efficient use of computational resources, recovery blocks have a decided advantage over N-version programming since with recovery blocks an alternative version of a program is executed only when a fault is detected. With N-version programming, however, all versions are executed each time a program is called. There appears to be little difference between the two approaches in terms of storage requirements. There is another criterion, in addition to efficiency, that distinguishes the two approaches. In the recovery-block approach the output of a program is subjected only to a generalized acceptance test, a test which in most cases will not embody total correctness. In N-version programming, on the other hand, the outputs of multiple versions of a program are compared which really amounts to a check on the total correctness of a computation. Backup software has the advantage of operating independently of the primary software so that problems do not arise from having to closely coordinate multiple software versions.

The N-version programming and recovery-block techniques have seen relatively little use to date. Avizienis has used student programs successfully to illustrate the application of N-version programming, but the examples are not representative in either complexity or style of the real-time software needed within AIPS. Gmeiner [3] and Taylor [4] have described the use of the N-version technique by the Swedish State Railways and in Germany for a fast-breeder nuclear reactor. Similar use experience with the recovery-block approach has not been found as of this writing. Because use experience has been limited, the cost-effectiveness of using these techniques has not been clearly established.

In summary, the various fault-tolerant software techniques are potentially useful for AIPS to improve software reliability. The AIPS design should incorporate features to facilitate the use of fault-tolerant software, where such features can be identified. However, it remains to be shown just how effective these techniques are, and whether their use can be justified on a cost basis.

3.3.2.3 Distributed Operating Systems

The distributed operating system for AIPS is responsible for providing a variety of functions, the principal ones of which are listed in the Software Survey viewgraphs of Appendix D. Existing distributed operating systems - a representative listing of which is also given in Appendix D, provide some of these functions; but no existing system provides them all, and moreover no existing system is designed to operate with stringent real-time constraints.

Two important issues in the design of distributed operating systems have been identified, namely:

Coordinating access to shared objects, and

Maintaining the consistency of data in the face of user errors, application errors, or partial system failure.

"Locking", "timestamps", "permits", and "tickets" have been proposed in the literature to solve the first problem, and each provides a method for ensuring that a particular correctness criterion - called "serializability" - is preserved. Various implementations of a software-structuring technique called "atomic actions" have been proposed to solve the second problem.

As a general result of our survey, we have learned - not unexpectedly - that the development and testing of distributing operating systems is "horrendously difficult". A distributed operating system for the real-time environment expected in the AIPS system does not exist. In developing such a system, it may be possible to adapt an existing transaction-oriented operating system, or alternatively if Ada is to be used for the AIPS implementation language, make appropriate modifications to an Ada uniprocessor run-time package.

3.3.2.4 Software Tools and Methods

This segment of the survey was undertaken to identify software support tools and methods which could enhance product reliability, reduce development and maintenance costs, and increase programmer productivity. A list of tools and methods which were evaluated appears in the Software Survey viewgraphs in Appendix D. The items in this list represent a broad spectrum of tools and methods to be used over the entire software-development life cycle. Since information is readily available on tools to aid in actual code development, the tools of most interest in the survey generally fell into three categories:

(1) Requirements and Design Aids
(2) Testing and Verification Aids
(3) Management Aids

Although many of the tools and methods were viewed in isolation, their use in the context of an integrated software-development environment was envisioned. A discussion of the three categories follows:

(1) <u>Requirements and Design Aids</u> - Tools in this category are used during the early part of the development life cycle. Historical data supports the use of early automated aids which can:

Represent the system and software requirements, and software architecture design, in a database,

Provide documentation support,

Enforce the use of standards,

Aid in project planning,

Provide a medium for product performance and quality assessment, and

Aid in increasing management visibility.

Few automated requirements aids exist in the marketplace and those that do are generally in the research stage. Examples include TRW's SREM, RSL/REVS, PSL/PSA[3], and DARTS[4]. Mature methods which are not automated include NRL's requirements methodology and the structured-analysis approach .

Automated design aids are more mature and include Intermetrics' BYRON, CSDL's DARTS, and Ada in the context of a programming support environment.

(2) Test and Verification Aids - Tools in this category are used during the later parts of the development life cycle. Like requirements aids, these tools are also in the research stage. Programming environments like the UNIX Programmer's Workbench, and the Army's APSE, the Ada Language System (ALS) provide some tools in this category. Structured testing, a methodology without tool support, is also useful.

(3) Management Aids - Management aids provide support in planning and control, and improve management visibility into an evolving software product during a development effort. Software planning aids, used to estimate costs and schedules, are recently available. WICOMO, an automated tool developed at the Wang Institute, uses Barry Boehm's popular COCOMO model which has been applied on a variety of aerospace applications. Tools which measure product quality are generally at the research stage of development. DARTS provides this capability and the methodology used can be applied to Ada designs and code. Programming environments such as the UNIX Programmer's Workbench, and the Army's ALS provide good tool support in configuration management and other control tasks.

3.3.3 Hardware

The hardware technology survey was focused on five subtopics:

(1) Processor Instruction-Set Architecture
(2) Semiconductor Technology
(3) Data-Net Transmission Medium
(4) Power Distribution and Control
(5) Mass Memories

Discussion of the survey results in each of these areas follows.

---

3 Problem Statement Language/Problem Statement Analyzer (University of Michigan)

4 Design Aids For Real-Time Systems (CSDL)

3.3.3.1 Processor Instruction-Set Architecture

The technology survey of processor instruction-set architectures was narrowed down to the two current military specifications and the more popular of the commercial microprocessors. The two military specifications, MIL-STD-1750A and MIL-STD-1862, are released and are in varied stages of implementation at the present. MIL-STD-1750A is an Air Force 16-bit architecture while MIL-STD-1862 is an Army 32-bit architecture. 1750A is further along in the development cycle with production units expected to be easily available in 1984. 1862 has just completed the brassboard stage with production not expected until 1986. All implementations must still be validated for compliance to their respective ISA. As with any ISA the availability of development tools is sparse, but will generally improve as production of ISA implementations commences. An advantage of a military standard is that changes to the ISA are tightly controlled by the issuing agency and not likely to vary over long periods of time.

Commercial ISA's are continually evolving based on manufacturers' marketing strategies. They are totally the property of the manufacturer and potentially could be phased out. In general, development tools are widely available from multiple manufacturers. Most ISAs are available in chip, board and system implementations. The more popular chips are designed to operate with coprocessors to extend the hardware capabilities. For example, math coprocessors can perform math routines in microcode. All manufacturers tend to develop complementing I/O chips based on typical needs. Some of the more recent microprocessors, such as the Intel 432, have been designed to accommodate high-order languages, error checking and multiprocessing.

3.3.3.2 Semiconductor Technology

In the survey of the semiconductor technology, current and developing manufacturing processes were examined. Bi-polar and MOS technologies are mature and the environment that they are suitable for is known. Current trends among manufacturers appears to be a conversion of bi-polar and n-channel metal-oxide semiconductor (NMOS) devices to complementary metal-oxide semiconductor (CMOS). New CMOS processing, with narrower line widths, is starting to yield devices with densities aproaching those of NMOS. The incorporation of the new CMOS processing results in lower power dissipation, wider temperature range, high noise immunity,less stringent power supply requirements and propagation delays under 8 ns. There are presently direct replacements for transistor-transistor logic (TTL) and NMOS microprocessor chips. Industry trends indicate that many more devices will be converted in the near future.

Gallium arsenide devices are appearing and they promise very high speed, low power dissipation and a high radiation tolerance. Presently some chips exist but a large commitment from the semiconductor manufacturers has not materialized.

Very high speed integrated circuitry (VHSIC), with its promise of high density and high speed, is just starting to develop. Presently, device types have been defined by the DoD specifically for military applications. The availability of VHSIC technology in commercial devices will lag the

corresponding realization in military applications by approximately two years.

3.3.3.3 Data-Net Transmission Medium

The media examined during this survey were copper wire and fiber optics. Of the two, copper wire is the more mature. Its transmission and environmental characteristics are well known and the interconnection of cables is a simple task. however, wire adds considerable weight and acts as a transmitter and receiver of electromagnetic interference (EMI).

Fiber optics is a less mature technology; however, it provides complete electrical isolation between systems, is immune to electromagnetic interference, is lightweight, and can carry high-bandwidth signals. Problems with fiber optics mostly revolve around environmental concerns. some of the concerns are: change in transmission loss associated with exposure to radiation; the effects of shock and vibration on interconnections and cable alignment; and the effects of temperature. Studies of the use of wavelength-division multiplexing for simultaneous multiple users and formats, and possible dedicated channels, are continuing at Langley Research Center. At present some passive coupling devices are being marketed and additional devices can be expected for the future.

3.3.3.4 Power Distribution and Control

The technology survey revealed that there has been considerable work done on reliable, efficient power generation and conversion for aerospace vehicles. Far less has been done on the design of power distribution and control systems having fault tolerance and redundancy. It is recognized that an autonomous method of power measurement and control is needed for projects where diagnosis and reconfiguration is extremely difficult and/or time consuming. Some attempts have been made to tailor a system for fault-tolerant operation. In the FTMP project there were quad power sources and a commutator at the load. The SIFT project also used quad power sources but diode "OR'd" power to the load. Aircraft use multiple independent power sources, independent power buses and battery backup for critical functions.

Since little has been done in the field of fault-tolerant power distribution and control, it is necessary for the AIPS program to study power distribution techniques, fault detection and recovery techniques and other problems associated with automated, highly reliable power systems.

3.3.3.5 Mass Memories

The mass memory media which was surveyed falls into four categories:

Magnetic tape,
Magnetic bubbles,
Discs, and
Semiconductors.

Each category is suitable for mass memory and can be ranked according to its speed and capacity. However, each category must be evaluated specif-

ically with regard to the mission requirements under which it will be used.

Although magnetic tape has a large capacity, it also has a long access time. Tape systems are subject to failures caused by the medium, such as head contamination and general mechanical wear.

Magnetic bubbles, with no mechanical components, are not subject to wear associated with reading and writing. Memory retention is performed by external permanent magnets. They are non-volatile and do not draw power in the standby mode. The availability of magnetic bubble memories has suffered in recent years due to the discontinuance of production by many manufacturers. The viability of bubbles cannot be assured at this time.

Discs are available in magnetic and optical versions. Optical devices have just recently become commercially available and it is too early to know about inherent problems.

Disc systems in general have problems with operation in severe environments and since they have rotating components are subject to wear and need of periodic maintenance.

Semiconductor memories have the fastest access time of the storage devices, and as densities increase building a mass memory in a small volume will be easily realizable. Semiconductor random-access memories (RAMs) are volatile and tend to require high power. Developments in CMOS RAMs with increases in speed and density, and the ability to maintain storage with low-power battery backup, now make them a potential candidate for mass memories.

3.3.4 Reliability and Performance Methods

The evaluation methodology subtopics (technical areas) surveyed were:

(1) Hardware Reliability, Maintainability, Availability (RMA)
(2) Software Reliability
(3) Computer System Performance
(4) Performability - A Combined Consideration of Performance and Reliability

These are discussed below in the order listed. Within each technical area, the survey sources are identified, the status of the technology within that area is assessed, and pertinent issues are raised. Conclusions are drawn for each area regarding the maturity of the technology and its suitability for application to the AIPS. The preferred measures of merit and evaluation approach are also discussed. Areas of outside support are identified in Section 4, to help overcome technology "shortfalls" which are evident as a result of the survey.

3.3.4.1 Hardware Reliability, Maintainability, Availability (RMA)

There are two major reliability modeling techniques - the combinatorial model and the Markov model. The technique with which most engi-

neers are familiar is the combinatorial method. The combinatorial reliability equations are constructed from a tree-structured reliability diagram, generally assuming a fault has occurred.

The mathematical basis for this technique is the successive application of a conditional probability equation

$$Q(SL) = Q(SL|E^1_1)\ P(E^1_1) + Q(SL|E^1_2)\ P(E^1_2) + \ldots$$

where

$Q(SL)$ = Probability of system loss

$Q(SL|E^1_1), Q(SL|E^1_2), \ldots$ = Conditional probability of system loss given the events E^1_1, E^1_2, ...

$E^1_1, E^1_2, \ldots$ = Events describing the status of component one

$P(E^1_1), P(E^1_2), \ldots$ = Probability of event $E^1_1, E^1_2, \ldots$

It is assumed that the set of events $(E^1_1, E^1_2, \ldots)$ are mutually exclusive and are collectively exhaustive.

An alternative method for computing the system reliability for large and complex systems is the Markov modeling technique. One of the principal reasons for using this method in fault-tolerant system reliability analysis is that it easily accommodates the incorporation of the fault-detection and isolation (FDI) decision errors as well as the FDI strategy into the reliability or availability model. The Markov model consists of a set of operational states for the system, a definition of the probability for transitioning between these states due to component failures or erroneous redundancy management system decisions, and the propagation of these probabilities in time.

It is interesting to contrast the Markov model with the combinatorial model. The Markov model is a microscopic approach to reliability modeling and the combinatorial model is a macroscopic approach. By this, it is meant that the Markov model focuses on events occurring over a single time step, whereas the combinatorial model focuses on the entire time interval. In classical combinatorial reliability modeling, it is usually assumed that the time order of a sequence of events occurring during the time interval of interest does not effect the reliability analysis. The Markov model, however, does account for the time order of events and in a manner which is transparent to the reliability modeler. Finally, the Markov model is a complete characterization of the system from a reliability point of view. The Markov model can contain far more information about the system that just the system unreliability. Because of the hierarchical tree structure and the top-down decomposition methodology used in the combinatorial model, it cannot contain the complete system characterization in a single model.

The viewgraph "Survey Sources", in the "RMA Summary" portion of Appendix D, shows the sources contacted during the RMA survey, the subject of discussion, and the mode of contact. The report by W. Ness of the Lock-

heed Georgia Co. and others [5] was the point of departure for this survey. It is an evaluation of reliability and failure-effect prediction methods for digital flight control and avionics systems. In all, twelve reliability-prediction methods and one failure-effects method, the fault-tree, were considered. Included were CARSRA, CARE III, ARIES, CAST, and TASRA, among others[5].

A great deal of attention was focused on CARE III. It is a general purpose reliability analysis and design tool for ultra-reliable fault-tolerant systems. It is a fault-handling Markov model based upon a probabilistic description of the detection, isolation and recovery mechanisms. A variety of fault and error models, both stationary and nonstationary, are included such as permanent, transient, intermittent, design faults, software errors and latent faults. As indicated in Figure 3-1 on page 3-23, CARE III has built upon the foundation laid by previous evaluation methodologies.

Dr. J. Stiffler, the developer of CARE III, was not available for consultation. However, valuable insight into the development and application of this reliability evaluation method was gained from a discussion with Mr. S. Bavuso, the technical monitor of the CARE III program [6]. CARE III is currently undergoing an independent evaluation by the Boeing Computer Services Co. Their goal is to review the mathematics and code of CARE III and "stress-test" the model. A user-friendly interface is also being developed for use with CARE III on the VAX computer system by the Research Triangle Institute. The release of CARE III is imminent. The subject of the generation of failure input data for CARE III, which may be obtained offline via simulation or testing, was also discussed with Mr. Bavuso. Information in this regard was also provided by J. G. McGough et al [7].

Drs. K. Trivedi and R. Geist of Duke University were also contacted during this survey [8], one reason being that Trivedi was a member of a CARE III Peer Review Group [9]. They are also responsible for HARP[6] which has potential application to AIPS.

The goal associated with the HARP effort is to develop a flexible reliability evaluation package which will aid in system design. It allows the user to determine the sensitivity of reliability to system parameters automatically and find the major drivers of reliability so that the system can be designed to optimize reliability. The sensitivity of reliability to initial state uncertainty, failure rates and coverage parameters can be assessed. HARP is applicable to large state-space systems by using the principle of behavioral decomposition just as in CARE III. That is, the

---

5 CARSRA --- Computer-Aided Redundant System Reliability Analysis (Boeing)
CARE III - Computer-Aided Reliability Estimation III (Raytheon)
ARIES ---- Automated Reliability Interactive Estimation System (UCLA)
CAST ----- Complementary Analytic Simulative Technique (Ultrasystems)
TASRA ---- Tabular System Reliability Analysis (Battelle)

6 Hybrid Automated Reliability Predictor (Duke)

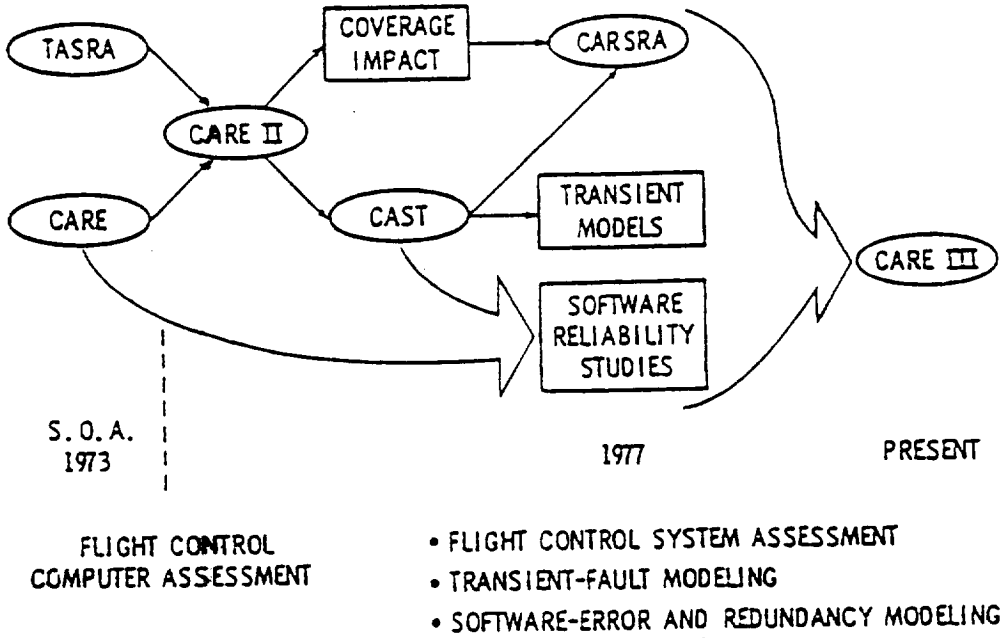


Figure 3-1. Technological Development Leading to CARE III.

fault-occurrence and fault-handling aspects of a computer system can be modeled separately because of the different frequencies at which they occur. This is done to optimize the performance of HARP.

Non-constant failure rates can be modeled in HARP using non-homogeneous Markov models. There is some limited evidence in the literature that this characteristic may be true for certain aspects of computer systems. Another characteristic of HARP is the use of Petri nets in the fault-handling model. A Monte Carlo simulation of this net is performed to get the probabilities of detection, isolation, etc., for determining the systems performance. This data is currently obtained offline for the CARE III model.

CSDL's MARK1 (not an acronym), a general set of modeling tools to help evaluate the reliability of complex systems, was also investigated as part of the RMA survey. The tools include a procedure for partitioning the system into subsystems, a method for developing models for subsystems from Failure Modes and Effects Analysis (FMEA) tables, and a specification language for translating the model description into an input file for a computer program. The computer program then numerically solves the specified models of subsystems, and combines the results from these models to obtain overall system reliability and other reliability parameters of interest. These tools are sufficiently general so that they will adequately handle diverse user-defined system architectures and are not limited to a predefined library of architectures.

A visit was also made to the Carnegie Mellon University to discuss transient fault data for RMA evaluation models [10] [11]. Battelle was contacted regarding a comparison they had made of three approaches to system reliability evaluation. E. F. Hitt, et al [12] provide a comparison on the TASRA evaluation modes, the fault-tree approach, and the performability- evaluation approaches. The Boeing Military Airplane Co. was contacted regarding their Stage-State reliability analysis technique, a fault-tree hybrid approach to the problem [13].

The literature, particularly the IEEE Transactions and the Proceedings of the International Symposia on Fault-Tolerant Computing, proved to be invaluable sources of information on RMA evaluation methodologies. Ng and Avizienis [14] provided information on ARIES while Costes, et al [15] was the source of information on SURF[7].

The reliability analysis of communication networks is one subject which will receive high priority during the AIPS program. This is due to the fact that networks may be used to a large extent in the AIPS architecture. The IEEE Transactions were surveyed to determine state-of-the-art approaches for defining, analyzing and evaluating networks, in general, and their reliability in particular.

Based on the survey conducted for the AIPS program (see the viewgraph "Status", in the RMA methodologies survey portion of Appendix D), it can be concluded that the RMA evaluation methodology technology is very mature. The Markov and fault tree approaches have been thoroughly studied by a number of organizations and several computer mechanizations are available for potential application to AIPS. Some comparison studies have been done among the various approaches and mechanizations to provide a better understanding of them and their relationship. The application of the mechanizations to practical problems has also been demonstrated with the results of these efforts interspersed throughout the literature.

There are several key issues which must be addressed in the selection of an RMA evaluation methodology for the AIPS program.

- One is the degree of automation in the mechanization of the RMA evaluation algorithm. The concern is that the knowledge required of the user about the system being evaluated may diminish with an increase in the degree of automation. It is felt that the user must know the system very intimately to perform a meaningful RMA evaluation.

- Another is the choice of the evaluation approach itself. Both approaches - the Markov model and the fault-tree combinatorial model - are valid. The selection of the approach for application must be based upon issues such as convenience to the user, availability, suitability to the task, etc.

---

7 (Not an acronym); Laboratoire d'Automatique et d'Analysedes Systems, Toulouse, France (LAAS).

- Failure and FDI system error models must also be addressed in the selection of an evaluation methodology and mechanization. The concerns in this area are whether or not the models are well defined, and whether or not the parameter values associated with them are supported by experimental data.

- Finally, the evaluation of communication networks is an issue to be resolved. Networks may be used in the AIPS architecture, so the mechanization selected for evaluation must be capable of accounting for them.

The conclusions drawn from a consideration of the hardware RMA evaluation methodology survey are as follows:

- The preferred measure of merit for the AIPS system is the number of failures per million hours given the maintenance interval.

- The Markov model is selected as the preferred evaluation approach. It is the most widely applied in the computer mechanizations surveyed. Furthermore, it is more versatile than the fault-tree approach because of its complete characterization of the system from a reliability point of view. The Markov model is the only feasible approach for evaluating system availability. The major shortcoming of the Markov model, the potentially large number of states required to define the system's operation, can be overcome using existing techniques.

- The evaluation of network interconnection reliability is a high-priority item for the AIPS program. The main reason for this is that the computer interconnections will probably dominate the overall reliability of the system.

- CSDL's approach to evaluating the RMA of the AIPS architecture during system development will be to use the in-house hybrid approach, MARK1. There are several reasons for adopting this approach. One is that CSDL has broad experience in the application of this mechanization. The MARK1 has been verified by modeling architectures that were studied by other organizations such as NASA LaRC and LAAS with good agreement. Furthermore, the MARK1 is very flexible and easily adaptable to new designs. Another significant reason for selecting the MARK1 is that minimal external help is needed to use the program.

  In summary, the major conclusion arrived at as a result of the hardware RMA evaluation methodologies survey is that the technology is very mature and can be applied to aid in the development and evaluation of the AIPS system.

3.3.4.2 Software Reliability

The sources contacted during the software reliability evaluation technology survey are listed in Appendix D. The number of contacts made is far fewer than that for the Hardware RMA survey. This was due to the infancy of software reliability evaluation technology relative to its hardware counterpart. A primary source of information for this survey is

the results of a recent CSDL internal research and development program, documented by DeWolf, et al [16]. That program was concerned with both software reliability evaluation models and data gathering. The information from this study was supplemented by telephone calls to J. Musa of Bell Labs, and A. Goel of Syracuse University on the subject of reliability models, and to V. Basili of the University of Maryland on the data-gathering field.

General conclusions can be drawn regarding the status of software reliability evaluation as a result of this survey. Several reliability models have been suggested by people active in the field. The Markov model, discussed by DeWolf et al [16], and the execution-time model and the differential-fault model, both of which are described by J. D. Musa [17], are examples. However, the major drawback in the evaluation of system software reliability is the lack of data to substantiate these models. As a result of this lack of data, a widely accepted computer mechanization for software reliability evaluation does not exist. The application of existing mechanizations to realistic programs is, therefore, also limited. An even more basic problem is that few comparison studies among the models exists.

The major conclusion of this phase of the study with regard to the AIPS program is that software reliability evaluation technology requires more maturity before an absolute evaluation of the candidate architectures can be made. It may be possible to perform an evaluation on a relative basis, with some degree of confidence, using existing models and data.

3.3.4.3 Computer System Performance

There are two general approaches to the performance evaluation of computer systems; analytic and simulation. Two general analytic options exist. One is the stochastic approach as defined by queuing theory and the other is a semi-deterministic approach in which randomness is dealt with using the mean values of the system random variables. Semi-deterministic analysis is based on the use of delay models as a function of system throughput.

The survey of the state of computer system performance evaluation (See "Sources," "Status," etc. in Appendix D.) focused on each of the approaches described above. The field is mature enough so that many textbooks have been written in the area. Several references were consulted, of which two focus on analytic approaches in general [18], [19] and a third is devoted strictly to the semi-deterministic approach [20]. Dr. K. Trivedi was also contacted [8] with regard to computer system performance evaluation as well as RMA evaluation as discussed previously. There was a decided emphasis on simulation with the other survey sources. Dr. Z. Segall discussed performance evaluation via simulation in general [11]. A visit was made to BMDATC to obtain information on their Test Bed. The Naval Air Development Center (NADC) was contacted regarding their General Computer System Simulation (GCSS) II performance evaluation tool. The technical literature provided supplementary information on all methods of computer system performance evaluation.

The focus of computer system performance evaluation technology has been on transaction-type systems rather than real-time applications, and the bulk of the information available is addressed to systems of this type. Several practical problems have been solved for transaction-type systems. A limited number of mechanizations for the evaluation of computer system performance are available. Comparisons between the different evaluation approaches are done on a heuristic basis only, rather than on hard factual data.

The status of computer system performance evaluation with regard to AIPS lies between the relative maturity of that for RMA evaluation and the immaturity of the state of the art of software reliability evaluation. Analytic approaches are preferred for the initial evaluation of the AIPS architectures due to the general structure and definition of the architectures at this time and the fact that they are more readily available for in-house use. The semi-deterministic approach is most applicable to architectures that have nearly deterministic workloads since the randomness is dealt with using mean values. Queuing theory requires a characterization of the workload statistics. A mix of these two approaches will be used for the AIPS. In the evaluation of the AIPS architectures the preferred measures of merit are delay time for each of the tools, efficient use of resources, and % overhead. These measures were selected based on the survey as well as the consideration of the definition of the requirements for the AIPS systems.

3.3.4.4 Performability - A Combined Consideration of Performance and Reliability

Performability is the measure of merit defined by the combined consideration of system reliability and performance. This subject has been investigated on a rather limited basis in the technical community as evidenced by the performability survey sources listed in Appendix D. CSDL applied this approach in evaluating the redundancy management system for the Inertial Upper Stage [21] . In parallel with this effort, and on an independent and more extensive basis, Prof. J. Meyer was applying this concept to computer systems [22]. Battelle Laboratories performed a comparison of performability, fault-tree analysis and the TASRA RMA evaluation approach [12]. Basically, all three approaches provide the same results for the examples considered.

Performability is a subject that is in the infant stages of its development. Only the basic ideas have been defined thus far, and its application to practical systems is very limited. Only one mechanization existed at the time of the survey.

Performability is viewed as the proper measure of merit for fault-tolerant systems because of its combined consideration of both reliability and performance. It has not yet been defined to the point where it can easily be applied to the AIPS system. This is not a major cause for concern at this time since it is not necessary to apply the concept of performability during the early stages of system design. The individual considerations of reliability and performance are more important during the initial stages of the evaluation of a system. An effort will be made to define a performability evaluation methodology for application to AIPS in later phases of the program.

3.3.5 Verification and Validation Methods

The survey of V&V methods will be described under three subtopics:

(1) Survey Results
(2) Issues Raised
(3) What Was Learned

The discussion below follows the "V&V Survey" viewgraphs of Appendix D.

3.3.5.1 Survey Results

The first contact was in the form of a visit to Dr.K.Trivedi at Duke University. The essence of our discussions involved semi-Markov modeling of software systems which although appropriate for design analysis is not directly applicable to the AIPS V&V problem.

The next visit was to Carnegie Mellon University. Doug Jensen enlightened us on his concepts for decentralized operating systems. However, he had not yet considered how one would verify or validate such systems.

Bob Loesh and Tal Brady of JPL reviewed V&V as applied to the Galileo project. Their initial comment was that it was a "horrendous" problem. Schedule and dollar problems thwarted efforts to incrementally and separately test the hardware and software. As a result, their first testing began at the fully integrated system level and, as one might expect, nothing worked when power was applied. Compounding the testing difficulties was the fact that the system had been intentionally designed without test "hooks" in either the hardware or software. Fortuitously, one word occurred in the same location every telemetry frame, thus providing a coarse system synchronization point.

Discussion with Herb & Myron Hecht of SoHaR Inc., regarding fault-tolerant software and associated hardware aids, indicated the need for "hooks" and as many stack registers as practical, for address information just prior to the fault, to facilitate both the implementation of fault tolerance as well as the testing thereof.

Fault-tolerant software was the topic of discussion with Dr. Avizienis at UCLA. His specialty in this area is N-version programming. He offered that the N-version approach was inherently self-validating. We briefly discussed the "N-versionness" of the Space Shuttle primary and backup flight software, and Dr. Avizienis indicated an interest in performing a comparison of the two.

Gerry Popek of ICA discussed and demonstrated his distributed operating system "LOCUS". Regarding validation he said that the system was extremely difficult to test.

A visit to Boeing Commercial Airplane Company in Renton Washington resulted in a tour of their facility for certifying the 757/767 FMS (Flight Management System) and discussions with several people involved with that process. Some of the salient points were that LRU (Line Replaceable Unit) V&V was done by the vendors, and that the Blue Label test pro-

gram was very effective in working out design problems on prototype LRU's in parallel with resolving requirements problems. The V&V of the FMC (Flight Management Computer) was fragmented. A small percentage of the code which was deterministic in nature was truly verified and validated. The majority of the code associated with displays and controls was considered validated through its use in CAB simulations.

Bill McDonald and his associates led us through a description of the BMDATC testbed, provided a tour of the facility, and demonstrated its operation. V&V is accomplished on this system by multiple runs (feasible since its problem-state time is small), automated fault injection, and statistical analysis. The associated operating system is verified by use (i.e. multiple runs).

Telephone conversations with John Hanaway and Dave Brown of Intermetrics, Seattle, Washington, regarding their involvement in the 757/767 V&V revealed that perhaps other people than those visited should be contacted due to their more direct involvement. Dave indicated that in his opinion the Boeing Blue Label Program was a very effective approach, and that the most significant problem area was that of getting the requirements correct.

The CSDL has experience on many previous and current programs. The most well-known of these was Project Apollo. The CSDL role in Apollo dealt with the primary guidance, navigation and flight control system hardware and software from concept to flight. This was a highly reliable single-string system achieved through careful design and intensive testing.

Space Shuttle involvement at the CSDL consisted of the following:

GN&C (Guidance Navigation and Control) program design for both the primary and backup flight systems,

Source code generation for the backup,

Comparison code generation for the primary,

Simulation of the vehicle and flight profiles,

Performance and common facility testing on the primary,

Unit-subsystem-integrated testing on the backup,

Simulation and analysis of the remote manipulator system, and

post-processing and analysis of flight data.

An independent V&V effort for the Defense Meteorological Satelite Program (DMSP) has been an ongoing program at the CSDL. This consists of receiving a package of flight code from the designer, testing it on a simulator and determining that it meets the system requirements.

The CSDL has designed, built, and demonstrated a Fault-Tolerant Processor (FTP).

The CSDL has participated in the F-8 digital fly-by-wire program from concept through flight. This is an ongoing program.

TRIDENT II is another ongoing program at the CSDL. Currently in the design stage it will continue on through flight testing. It involves a multi-microprocessor control system and the verification thereof.

The scope of the V&V effort with which this survey was concerned includes monitoring the design and development of hardware and software culminating with their integration into a system.

A diagram showing the relationship of V&V activities to software development is contained in the V&V viewgraphs of Appendix D.

Some examples of approaches to V&V at the system level are the following.

- Space Shuttle

  SAIL (Shuttle Avionics Integration Laboratory)
  - This facility is essentially an iron bird with a complete set of flight-configured hardware and software connected to a large simulator. Typically, it runs predefined segments of nominal and off-nominal mission profiles and is checkpoint restartable. It is large, expensive and has limited data insight.

  SVDS (Shuttle Vehicle Dynamics Simulator)
  - This facility evaluates the ability of flight avionics to cope with the modeled vehicle dynamics.

  SMS (Shuttle Mission Simulator) - This facility is essentially a crew-training device with a moving- base cockpit. It primarily stresses the crew interface with the flight software.

  FSL (Flight Systems Laboratory) - This Rockwell facility can be operated with or without a person in the loop for entry-through-landing simulations using a subset of flight hardware and complete flight software.

- 757/767

  CS&FCL (Computer Simulation and Flight Control Laboratory) - This Boeing facility can run various configurations of avionic LRUs in work-station-type environments with various levels of simulation up to and including moving-base man-in-the-loop cabs[8].

- BMDATC

[8] (not an acronym)

Testbed - This facility is a multinode, multicomputer test facility specifically designed to verify and validate the BMDATC crossbar which consists of multinoded fault-tolerant microprocessors.

- F-8

  Iron Bird - This facility is similar to the SAIL in that it replicates the full set of flight hardware and software for a tactical fighter aircraft. It uses primitive fault-injection techniques as part of the verification process.

- FTMP (Fault-Tolerant Multiprocessor)

  Verification of the FTMP was performed using **automated simulation and automated fault injection.** It is planned for AIRLAB to extend this testing.

Some examples of V&V techniques applied to hardware are the following:

- FTMP

  Two methods were applied to the FTMP for its verification. One was a **FMEA (Failure Modes and Effects Analysis)** effort in which the effects of predicted failures are determined and evaluated. The other technique was **automated fault injection.** This was accomplished by a computer-driven fault generator which injected the programmed faults into the circuitry through a special device-extender module. Using this technique, 20,000 faults were introduced into the FTMP and approximately 17,000 were properly detected and handled. The 3,000 undetected faults were determined to be "don't care" situations such as applying a ground signal to a pin already attached to ground.

- F-8

  FMEA/FMET(Failure Modes and Effects Test) - This FMEA is the same technique as described for the FTMP. The FMET is theoretically the same as the fault-injection scheme used on the FTMP except it was carried out manually, hence perhaps not as comprehensively.

Some examples of V&V techniques applied to software are the following:

- Shuttle

  SDL/SPF (Software Development Laboratory/Software Production Facility) - The SDL was used for the development of IBM coded software for the shuttle primary flight system. This effort included unit tests, function tests, subsystem tests and integrated system tests using appropriate drivers, stubs, as well as full-blown mission simulation. The SPF is an upgraded SDL which has the additional capability of performing software development and test for the backup system.

SLS (Statement-Level Simulator) - The SLS is a high-fidelity simulator developed by the Draper Laboratory. It was used to test the Draper version of the shuttle primary flight system (coded by Draper). The results of these tests were then compared to the same tests run at the SDL and other facilities like the Rockwell FSL (see below.) The SLS was also used for testing the integrated source code for the backup flight system. This code was developed by Draper, Rockwell, and Intermetrics on different facilities and integrated at the Draper facility.

FSL (Flight Systems Laboratory) - The FSL at Rockwell was used for development of a portion of the backup system code, as well as for entry-through-landing tests of both primary and backup flight code.

- DMSP

MSTF/DSS (Multiprocessor Software Test Facility/Digital Software Simulator) - The DMSP project used the MSTF/DSS work-station-like facility to verify and validate the flight code developed by RCA. The simulator is a high-fidelity, high-visibility, highly automated tool. It has the ability to validate unmodified flight code in a dynamic, closed-loop environment.

- TRIDENT I

The TRIDENT I software was developed and tested in an orderly fashion from the unit level to the integrated system level. Performance tests were accomplished on a digital simulator.

3.3.5.2 Issues Raised

- System

The issues to be addressed at the system level are:

(1) Can the reliability of the system be demonstrated?
(2) Does the system conform to the requirements which initiated the design?
(3) What is the performance of the system under stress conditions?
(4) Are the models used for development, test, verification and validation correct?

- Hardware

Hardware issues include the degrees of FMEA/FMET and automated fault injection, and the correctness of models used in the design.

- Software

In the world of software, some issues are:

(1) How much automated testing should be done and to what degree should it be automated?
(2) What kind of simulation/emulation is required and at what stage of development are they most effectively introduced?
(3) What tools and/or techniques are available or need to be developed?
(4) What level of effort of IV&V is sufficient?

- General

In all three areas - **system, hardware and software** - it is necessary to assure that sufficient "built-in Hooks" and "aids for testing" are included in the design to facilitate testing, documentation, and data collection and analysis. It also appears highly desirable to have some method of clock control for obtaining "snapshots" of the global state at predetermined points.

3.3.5.3 What Was Learned

During the course of this survey several things have become apparent:

Depending upon the language there seem to be many "development tools" available (with the exception of Ada).

There are many "techniques" applicable to V&V of fault-tolerant systems.

There are no portable, directly applicable "tools" for AIPS V&V.

Additionally, there are no tools or techniques for V&V of distributed or decentralized operating systems.

The failings of most V&V efforts appear to be:

Deficiencies in the requirements,

Schedule inequities,

Insufficient resources, and

Late implementation.

The best guidelines available today for verifying and validating digital avionics appear to be in the Radio Technical Commission for Aeronautics document, RTCA/DO-178.

SECTION 4

TECHNOLOGY SUPPORT

4.1 Introduction

This section pertains to technology areas which have been identified as being particularly relevant to the development of distributed fault-tolerant systems such as the AIPS, and for which further development would be fruitful. These areas are divided between a list of items recommended for development by CSDL in direct support of AIPS and a proposed list of related research topics for NASA consideration.

4.2 Recommended AIPS-Supporting Technology Development by CSDL

The technology and tool development proposed for CSDL development is that which has been judged to be the most supportive of AIPS development goals and is within the scope of the program.

4.2.1 High-Order Language Selection

Our recommendations, after doing the survey of high-order languages, are as follows:

- CSDL should acquire and gain experience with Ada compilers.
- An Ada program design language should be used in the design of the AIPS system.
- CSDL should, if possible, acquire an Ada Programming Support Environment (APSE) for software development.
- CSDL should acquire or develop a run-time environment for the AIPS system to support code development and testing in Ada.

In the event that the Ada compilers are not ready when it is time to begin coding, the design shall have been done using Ada constructs, and there are HOL's available in which we can do initial coding, no matter which processor is chosen.

4.2.2 Fault-Tolerant Software Experiment

Recovery blocks and N-version programming are two possible approaches to implementing fault-tolerant software. To aid in evaluating the effectiveness of each approach, it is proposed that an experiment be performed in which two programs are written, each incorporating one of these two techniques. The two programs would then be tested using fault-injection techniques to determine the fault coverage of each technique.

4.2.3 Software Reliability Data Gathering

A variety of software reliability models have been proposed in the literature. A subset of these models should be selected for evaluation on the AIPS program to provide estimates of software reliability. To support evaluation and use of these models, it is proposed that error data be gathered during AIPS software development and testing. These data would pertain to the frequency and category - e.g., clerical, design, specification - of errors uncovered. Data collection would be made as automatic as possible by using facilities provided within the software development environment.

4.2.4 Support Software Development

Support software for the AIPS encompasses those tools which will enhance product reliability, reduce development and maintenance costs, increase developer productivity, and enhance management control and visibility. The expected volume of software and its associated documentation, and the complexity of the AIPS, mandates the best available tool support. The AIPS software toolset should be an integrated package with a well-defined method for use and application during the development life cycle. To obtain this capability, it is proposed to start with an Ada Programming Support Environment if possible, and acquire or develop additional tools as needed. The tools that have been identified in the survey should form the basis for make or buy decisions. In addition to the advantages of using software tools listed previously, these tools represent a class of software that can be reused within many different environments.

4.2.5 Hardware Reliability, Maintainability and Availability(RMA)

The evaluation of network interconnection reliability is a high-priority item for the AIPS program. The main reason for this is that the computer interconnections will probably dominate the overall reliability of the system.

4.2.6 Computer System Performance

An evaluation of the GCSS II and the BMD Test Bed should be conducted, in parallel with the analytic evaluation of the AIPS architectures, to determine their suitability for use on the AIPS program in the evaluation of computer system performance.

4.2.7 Verification and Validation Development

Due to the iterative nature of the design process of sophisticated systems, documentation tends to have multiple revisions. Some effort should be made to automate document/report generation and revision. Additionally, testing should be considered strongly in the design process such that "hooks" are generously provided in both hardware and software to simplify the test and analysis necessary for validation.

4.3 Proposed Research Topics for NASA Development

The technology topics proposed for NASA R&D are discussed below. In general, they are of a more research-oriented nature and are less likely to be central to the AIPS development. However, they are expected to combine with AIPS in a synergistic fashion to produce tools and methods supportive of future system developments.

4.3.1 Distributed Processing Using Ada

Currently Ada compilers are targeted to single processors. There are issues that are not well understood surrounding the use of Ada in a distributed processor environment. The investigation of these issues is recommended as a suitable task for NASA.

4.3.2 Cost Effectiveness of Fault-Tolerant Software

N-version programming and recovery blocks are two techniques that have been proposed for improving the reliability of software. Questions have been raised, however, about the cost-effectiveness of these approaches to fault-tolerant software. In particular, for a given level of funding, can high reliability be best achieved by developing multiple, independent versions of a program or by developing a single, ultra-high-reliability version?

It is proposed that NASA develop a model for estimating the cost effectiveness of both the single-version and the multiple-version approaches. To accomplish this task, it will be necessary to model software reliability for fault-tolerant and non-fault-tolerant software, as well as to model software reliability improvement due to testing.

4.3.3 Correlated Errors Among Multiple Versions of Software

Both the N-version and the recovery-block approaches to software fault-tolerance rely upon a crucial assumption: that the errors (bugs) in independently developed versions of a program are uncorrelated, i.e., that they do not appear in all the versions. The validity of this assumption has been examined for N-version software at UCLA using student programs.

To test the validity of this assumption for operational programs, it is proposed that NASA examine empirical data from previous large programs (such as the Shuttle) in which multiple program versions were employed. European experience with fault-tolerant software should also be reviewed.

4.3.4 Taxonomy for Distributed Operating Systems

The character of a distributed operating system is determined by a number of relatively independent design choices. For example, What is the connection strategy? How do tasks or data migrate through the system? How is contention handled? What is the structure of the file system?

It is proposed in this task that NASA develop a taxonomy of distributed operating systems based on the possible choices for each of these design decisions. The taxonomy should be illustrated using examples of existing distributed avionics systems such as the Digital Avionics Information System (DAIS) (at AFWAL), FTMP (at AIRLAB), SIFT (at AIRLAB) and general purpose systems such as LOCUS (at UCLA) and Medusa and StarOS (at CMU).

4.3.5 Applications Sensitivity of Fault-Tolerant Software

It is proposed that NASA assess the use of various fault-tolerant software schemes such as N-version, recovery blocks, and food-taster. Their appropriateness should be evaluated for a range of avionics applications such as real- and non-real-time functions, functions which perform a portion of a complex control process (e.g., flight control), functions which have complex filtering and threshold data-selection decisions (e.g., navigation) and so forth.

Fault detection and isolation implementations for each of the considered fault-tolerance schemes including estimates of fault coverage and false-alarm probabilities should be defined and evaluated for each of the applications. Included would be a definition of the module or function level at which fault decisions are made.

Finally, this task would define the application characteristics that determine which of the fault-tolerance schemes is most appropriate.

4.3.6 Hardware Reliability, Maintainability and Availability (RMA)

Technical areas exist in which outside support could be applied to aid in the evaluation of AIPS-type systems. One major area is in the definition of system parameters and failure models. Evidence exists that the processors associated with the AIPS architecture may have non-constant failure rates [10] Additional data is needed to provide increased confidence in this area. Also, the parameters associated with the latent-fault model of CARE III are only partially defined by experimentation. Additional information is needed in this area.

Another area for outside support concerns an independent RMA assessment of the AIPS architecture. The results of this undertaking would pro-

vide increased confidence in the RMA assessment if the CSDL and outside sources agree. Areas requiring further investigation would be highlighted if the results differ.

Data is also needed with regard to the various processor/processor and processor/sensor interconnection techniques such as multiplex buses, ring buses, mesh, star, etc. It is suggested that a data base of user reliability and performance experience with the various interconnection configurations be constructed and that the data be used to modify (or validate) reliability models to represent actual experience. Furthermore, it is proposed that an active monitoring of failures be made for updating the data base. In addition, it is further proposed that specific tests be designed and implemented on the various NASA test beds to further characterize performance of the interconnection techniques (e.g., transport lag, reconfiguration time, throughput).

4.3.7 Software Reliability Evaluation

Software reliability evaluation is a fruitful area for the application of outside support to enhance existing technology. One suggested area for support is the collection of data to support existing software reliability evaluation models. A more encompassing and ambitious task would be to develop a methodology for the software reliability evaluation of real-time redundant, distributed systems to a level commensurate with the highly reliable systems of the future. This task would encompass model development, evaluation and comparison, and the definition of the model environment in terms of a definition of module reliability based on size and complexity, software reliability improvement due to validation and verification, testing, etc. The goal of this effort should be the definition of an adequate software reliability prediction mechanism.

4.3.8 Computer System Performance Evaluation

The major area in which outside support can be applied to aid AIPS and similar programs is in the development of a performance evaluation methodology for real-time redundant, distributed systems. This methodology could then be used to make fundamental comparisons among competing architectures as well as more detailed evaluations during the design and development phases of a specific system.

4.3.9 Performability Evaluation

A potential area for outside support would be the development of a performability applications methodology for real-time, redundant, distributed computer systems. This methodology should be applicable to the evaluation of the spectrum of NASA applications. Issues which must be addressed in the development of this methodology include defining performance and reliability measures for the different applications, defining performability goals, the development of computer programs for evaluation, the integration of system hardware and software reliability evaluation into the performability evaluation and the confirmation of results with experimental data.

4.3.10 Verification and Validation Development

Two major tools need to be developed. One is a practical, usable design language to automate the "requirements--->specification--->code" chain. The other is a realizable tool or technique for validation of decentralized operating systems.

APPENDIX A

INTERACTION REPORTS

This appendix contains internal memoranda in which certain interactions with outside sources of relevant technology information were documented for the record during the AIPS Technology Survey.

Two blocks of memos are included, namely, AIPS memos and ADA memos. The memos are presented in numerical order (which means, also, approximately in date order) within each of these blocks. The AIPS memos appear first, then the ADA memos.

The ADA memos pertain to the high-level programming language Ada. Ada was designed and developed for the DoD for use in embedded systems. Ada is a registered trademark of the U. S. Government (Ada Joint Program Office).

The Ada-Jovial Users' Group (AdaJUG) is a group of personnel from DoD contractors and companies which produce or use JOVIAL and/or Ada compilers and tools. Before Ada, JOVIAL was the required language for use on all Air Force embedded computer systems software.

AdaTec is the Technical Committee for Ada within the Special Interest Group for Languages (SIGPLAN) in the Association for Computing Machinery (ACM).

For more information pertaining to Ada, contact the Ada Joint Program Office (AJPO), Room 3D139, (400 A/N) Pentagon, Washington, D. C. 20301, telephone (202) 694-0209.

(The foregoing Ada-related acknowledgements have been deleted from individual memos before their incorporation in this Appendix.)

(Text of CSDL Internal Memorandum)

To: R. O'Donnell Memo No: AIPS-83-35

From: P. Szulewski

Date: 15 June 1983

Subject: Trip Report - NASA Langley

Copies: Distribution

The following trip report is written to conform with the Template for Software Technology Survey, which was identified in AIPS-83-24.

1. Topic

Software Tools and Computer Science Research at LaRC

2. Background/History

On 2 June, P. Szulewski met with Susan Voigt of LaRC. She is a key person in LaRC's Aerospace Computer Science Research Program and has just completed an assignment with all NASA centers which produced an intercenter Computer Science Research Program Plan (NASA Tech. Memo. 85631).

During the visit, Susan and I discussed a variety of issues, ranging from general NASA Computer Science Research goals to specific tools being developed within her division.

3. Technology Description

The following subsections describe the technology that we discussed. It generally falls into the area of Software Engineering Tools and Methods.

3.1 Aerospace Computer Science Research at LaRC

LaRC is presently sponsoring a continuing program in Computer Science Research which they feel is relevant to advanced development, primaririly the space station. There are five broad areas of concentration.

- Concurrent Processing

- Software Engineering
- Fault-Tolerant Software
- Information Management
- Computational Mathematics

These areas are outlined in brief detail by the viewgraphs attached. Additional viewgraphs provided are "unofficial" milestones and targets for some of the projects underway in the Analysis and Computation Division (ACD) and Flight Control Systems Division (FCSD).

3.2 NASA Computer Science Research Plan

In March 1983, NASA released their Computer Science Research Plan. Susan Voigt chaired the responsible committee. The plan was constructed to focus on broad issues which are either critical or unique to NASA's mission. The resulting program concentrates on three themes.

- Concurrent Processing
- Highly Reliable Cost-Effective Computing
- Scientific and Engineering Information Management

The plan is very detailed and is available from the author of this memo (see Reference 1).

3.3 HAL/S Integrated Verification and Testing System (IVTS)

I was given a demonstration of the IVTS during my visit. This system is presently under development and many of its features are operational. IVTS is an interractive tool used to verify and test programs written in HAL/S. It provides static analysis, symbolic execution, dynamic analysis (testing), and documentation support. These features are described in Reference 2, which is available from the author of this memo.

3.4 Software Development Environments

LaRC has just purchased and is expecting delivery this month of a commercially available Software Development Workstation. This Workstation is 68000 based and runs as a standalone processor which is expected to be networked into their mainframe (CDC Cyber). The operating system and tools are supposed to be Unix or Unix-like.

4. Relevance to AIPS

All of the technology issues discussed were relevant to AIPS since AIPS is a fallout of the overall NASA research plan. In particular, the IVTS and workstation experience may be two things to keep track of. If AIPS uses HAL/S, the IVTS may be useful during program development. Unix workstations are presently catching-on with the industry. Although Unix is not used presently at CSDL, it will be soon and it appears to be very good.

User Experience

LaRC is presently experimenting with tools and workstations with a very small staff. Over the next year they should have some results of interest.

Status

See viewgraphs on milestones attached.

***** EDITORIAL NOTE *****:

The viewgraphs referred to were omitted in this appendix

Future Prospects

It is expected that once some results are available, we could check in again for an update.

Conclusions/Recommendations

LaRC is experimenting with tools that we might be interested in when we begin software development. The tools they are using are HAL/S and Pascal oriented. Their experience could be helpful if we chose to use HAL/S.

References

(1) "NASA Computer Science Research Plan," NASA Technical Memorandum 85631, March 1983.

(2) Senn, E.H., Ames, K.R., Smith, K.A., "Integrated Verification and Testing System (IVTS) for HAL/S Programs," to be presented at Softfair '83, July 1983.

(Text of CSDL Internal Memorandum)

To: R. O'Donnell Memo No: AIPS-83-42

From: M. Whalen

Date: July 8th, 1983

Subject: Visit to Intermetrics, Inc. to Discuss Ada

Copies: AIPS Distribution

TOPIC:

Ada

BACKGROUND/HISTORY

The Department of Defense (DoD) has designated Ada as the DoD's standard language for embedded systems and is dedicating a large quantity of resources to its development. In an attempt to reduce the costs associated with the development, testing, and maintenance of the software for embedded systems, the DoD has committed an enormous effort to the design of the Ada language specification and the Ada Program Support Environment (APSE).

Intermetrics has a contract with the Air Force to develop an Ada environment for the IBM 370 series of computers with the VM operating system.

On Friday, May 20 1983 Paul Szulewski, Roger Racine and I went to Intermetrics, Inc. to talk with Mike Ryer about Ada. Mike is in charge of Ada at Intermetrics. For a long time he was involved with the Hal language, and worked with some of us in the early days of the STS Backup Flight System (BFS), so he is very familiar with CSDL, and the work and people in the NASA Department. He was very candid and helpful in his discussion of where they were in the development of their Ada environment.

TECHNOLOGY DESCRIPTION

Ada is a High Order Language designed and developed for the DoD for use in embedded systems.

RELEVANCE TO AIPS

A requirement of the AIPS program is that the software for the program be done in a High Order Language (HOL). The general nature of the AIPS

program, the fact that we are looking to future technologies and the possible application to DoD programs , makes Ada a candidate to be that HOL.

USER EXPERIENCE

The people at Intermetrics have written hundreds of thousands of lines of code in Ada. They have been involved with Ada since its inception. We discussed the training of those people in Ada, and Mike pointed out that they had ten years of experience with the language that included involvement in its design. Essentially, they started with a core of people who knew Ada and new people have been able to use that core as tutors. They have some in-house courses but he feels that hands on experience is more useful. Most of the people at Intermetrics and, in fact everywhere, I think, that are currently using Ada are compiler writers. This is an entirely different breed than the application specialists that we have need for in programs like AIPS. Mike feels that training his compiler writers to be competent Ada programmers takes a minimum of six months. We agreed that training engineers in the mind set of Ada may take even longer.

STATUS

The Intermetrics compiler is scheduled to be delivered to the Air Force in November of 1983 and Mike maintains that they will meet that schedule. The environment is scheduled for six months later and they are on schedule there also. By delivered, they mean they will have passed the Ada Joint Program Office (AJPO) validation tests and will be ready for Beta testing. This compiler and environment are being developed for the IBM VM system which is not the operating system that we currently have on the Amdahl. We do have VM resident on the IBM 4341.

They have not yet signed a contract to build an Ada compiler with a 1750A architecture but expect to do so this summer.

Intermetrics is not building runtime packages, but expects that they will be available through the AJPO.

The compiler that is delivered in November will not produce optimized code, but an optimizing compiler is due from Intermetrics six months later.

CONCLUSIONS/RECOMMENDATIONS

It is too soon to decide whether to select Ada as the implementation language for the AIPS program. The Intermetrics schedules seem to be intact, but even if delivered on time, it is a new language with concepts that are unknown to most of us, and it will be untested in any field use. However, I think that we should use an Ada program design language, of which there are several available, and we should take care to design our software in such a way that a conversion to Ada in the future is not impossible.

(Text of CSDL Internal Memorandum)

To: R. O'Donnell Memo No: AIPS-83-43

From: M. Whalen

Date: July 8th, 1983

Subject: Call From Softech About Ada

Copies: AIPS Distribution

TOPIC:

Ada

BACKGROUND/HISTORY

The Department of Defense (DoD) has designated Ada as the DoD's standard language for embedded systems and is dedicating a large quantity of resources to its development. In an attempt to reduce the costs associated with the development, testing, and maintenance of the software for embedded systems, the DoD has committed an enormous effort to the design of the Ada language specification and the Ada Program Support Environment (APSE).

Softech has a contract with the Army to develop an Ada environment for the VAX system.

After several attempts to set up a meeting with the people at Softech to discuss their Ada development, Mr. Jim Larue from there called me on Thursday afternoon, July 7, 1983. We have tentatively set up a meeting at Softech for Tuesday July 26, 1983 at 9:30 a.m. Mr. Larue was very helpful once he understood the nature of the questions that we have about Ada, and we spent almost an hour on the phone talking about various Ada isssues. I will summarize our conversation in this memo.

TECHNOLOGY DESCRIPTION

Ada is a High Order Language designed and developed for the DoD for use in embedded systems.

RELEVANCE TO AIPS

A requirement of the AIPS program is that the software for the program be done in a High Order Language (HOL). The general nature of the AIPS

program, the fact that we are looking to future technologies and the possible application to DoD programs , makes Ada a candidate to be that HOL.

USER EXPERIENCE

Softech has a contract to build Ada systems on the VAX host with two targets: the VAX, and an unspecified computer with a NEBULA architecture. The interim compiler is scheduled to be delivered to the Army in October of this year with the production compiler to be delivered in July, 1984. They have another contract with a different agency, (he didn't say which one), to build Ada for an 8086 target. That compiler is also scheduled for July, 1984. They had expected to get a sole source contract for a 1750A Ada, as had Intermetrics, but the goverment has decided to go out to bid for it. The RFP is out but the contract has not been let. He says that the Softech environment has been finished and that they are now 'tuning' the compiler. The environment has been written in Ada, but is then put through a Pascal translator, and compiled on a Pascal compiler. The Pascal compiler produces inefficient code. I asked if 'tuning' means optimizing, and he informed me that they have two optimizers, one local and one global. We talked some about education of users, and he was especially helpful in this area. Although he didn't have the answers, he gave me the names of people in other companies who have used Ada to write application software and suggested that I contact them using him as a reference, and discuss their experience with them.

STATUS

I think I have covered the status as garnered from this phone call in the above section. After the July 26 meeting I may have some harder data.

CONCLUSIONS/RECOMMENDATIONS

We shall take a group to Softech on July 26 and after that we will have conclusions/recommendations. I shall also call:

Mr. Sterling McCullogh, (312)933-5290 of Sonicraft, Inc. (They have developed software for the Electronic Systems Division (ESD), Air Force Systems Command (AFSC), using Ada. He is the head of their software engineering group);

Ms. Charlene Haydyn, (617)449-2000 of GTE; and

Ms. Carol Rughini, (617)366-6000 x4535 also of GTE.

(Haydyn and Rughini are both involved with projects that have used Ada at GTE.)

(Text of CSDL Internal Memorandum)

To: R. O'Donnell Memo No: AIPS-83-51

From: B. DeWolf, R. Werner, and P. Motyka

Date: August 2nd, 1983

Copies: See Distribution

This visit was arranged through the efforts of Dan Siewiorek, although he was unable to meet with us himself due to travel commitments. We met with Zary Segall, Mahadev Satyanarayanan (Satya), and Doug Jensen.

1. MEETING WITH ZARY SEGALL

Topic

Distributed Systems and Evaluation Models

Background/History

Carnegie-Mellon has had a long involvement in distributed computing system development and modeling analysis. The Cm* 50-processor multiprocessor is a current machine with roots in previous multiprocessors such as the 16-processor C.mmp. The SPICE project is a current distributed computer implementation with a message-based operating system.

Technology Description

Segall's current interests center around environments for (1) designing, (2) building, and (3) evaluating distributed systems. They are attempting to match algorithms and architectures by empirically determining performance using distributed test beds and the three environments mentioned above. They are doing work for both RCA/AEGIS and BMD (Bill McDonald at System Development Corporation (SDC) in Huntsville) in evaluating distributed designs. Although the emphasis is on empirical evaluation, they do make use of reliability analysis tools. Tools mentioned included Advisor (a PMS-based combinatorial model), Lambda, Relicalc, and Stars. Their work is partly motivated by the observation that performance on previous multimicroprocessors has not been significantly better than current uniprocessor performance. They are attempting to build their development environments using a relational data base and an artificial intelligence (AI)-oriented design-aid tool to enable the designer to find answers to high-level questions concerning system performance (bus loading, bottlenecks, queue statistics, etc.).

Relevance to AIPS

Segall's system will probably not have any direct use on AIPS since it's in too early a development stage, but the pointer to the work going on at BMD is most useful and should be followed up. The reliability models are state-of-the-art, but probably don't provide any capabilities not currently present in-house.

User Experience

For the testbed, developers only and limited (see references). Considerable experience with the reliability models.

Status

Portions of the system exist in prototype form.

Future Prospects

The project is ambitious and experimental in nature. If successful, it could be quite useful to AIPS.

Conclusions/Recommendations

The goals of the project point up a serious problem for AIPS: how to tune algorithms and architectures, and how to get meaningful performance measurements based on a prototype of the system. The BMD testbed should be explored as a possible way of empirically evaluating certain distributed approaches.

References (copies available from B. DeWolf)

1. York, G., Siewiorek, D., and Segall, Z., "Asynchronous Software Voting in NMR Computer Structures," 31 March 1983, preprint.

2. Segall, Z., et. al., "An Integrated Instrumentation Environment for Multiprocessors," IEEE Trans. on Computers, Vol. C-32, No. 1, Jan. 1983, pp. 4-14.

3. Wilson, A., Siewiorek, D., and Segall, Z., "Evaluation of Multiprocessor Interconnect Structures with the Cm* Testbed," 1983, preprint.

4. Singh, A., and Segall, Z., "Synthetic Workload Generation for Experimentation with Multiprocessors," Proc. 3rd. Int. Conf. on Distributed Computing Systems, Oct. 1982, pp. 778-785.

II. MEETING WITH MAHADEV SATYANARAYANAN (SATYA)

Topic

Carnegie-Mellon Local Area Network

Background/History

Carnegie-Mellon has embarked on an ambitious program to provide a campus-wide distributed network of personal computers for CMU students.

Technology Description

The project is a joint CMU/IBM effort to start as a 20 to 30 node system, and grow to 5000 nodes. Much emphasis is being placed on resource sharing for file and program storage, while making the network essentially transparent to users. An ethernet broadband communications medium is anticipated. Terminals will be grouped into clusters which will be responsible for file storage and access services. The network operating system will be a collection of communicating local operating systems. All communication is message based.

Relevance to AIPS

None at present.

User Experience

None.

Status

In planning and early design phase.

Future Prospects

Distributed operating system may be of some interest.

Conclusions/Recommendations

The project is in too early a phase to be of current interest.

References

Satya has published an often-cited reference book on multiprocessors (Prentice-Hall).

III MEETING WITH DOUG JENSEN

Topic

Decentralized Control

Background/History

Since coming to CMU from Honeywell about five years ago, Jensen has been concerned with the decentralized control of distributed computer systems.

Technology Description

Jensen's current work deals mainly with conceptual and philosophical issues related to decentralized resource management, although a 5-work-station network is planned (using ethernet and Unix) to allow experimentation. Jensen's contention is that resource management in most current distributed systems is centralized to a great degree, relying on a shared knowledge of the system state. Cooperating managers in such a system must arrive at some consensus view of the state, despite failures, communication delays, and configuration changes. This results in high tolerance. Jensen's approach in his so-called "Archons" project is built on several elements that are not yet fully explored. These include: decision making on a probabilistic basis in the absense of a complete knowledge of system state, elimination of unnecessary protocols and synchronism, the serialization model of data communications, and distributed problem solving techniques from AI.

Relevance to AIPS

Nothing concrete at present, although Jensen's papers are lucid and insightful and well worth reading.

User Experience

None.

Status

Mostly in conceptual phase.

Future Prospects

Likely worthwhile contributions to a host of fundamental issues in decentralized control.

Conclusions/Recommendations

The work should be supported for the sake of future insights it is likely to provide, but its near-term relevance to AIPS is limited.

References (copies available from B. DeWolf)

1. Jensen, E.D., "Decentralized Executive Control of Computers," Proc. 3rd. Int. Conf. on Distributed Computing Systems, Oct. 1982, pp. 31-35.

2. Lehoczky, J.P., et. al., "Scheduling Non-Serializable Transactions in Decentralized Operating Systems," 21 Feb. 1983, preprint.

3. Colwill, R.P., Hitchcock, C.Y., III, and Jensen, E.D., "Peering Through the RISC/CISC Fog," March 1983, preprint.

4. Sha, L., et. al., "Distributed Co-Operating Processes and Transactions," 1983, preprint.

5. Jensen, E.D., "The Archons Project: An Overview," March 1983, vugraphs.

(Text of CSDL Internal Memorandum)

To: R. O'Donnell Memo No: AIPS-83-56

From: P. Szulewski, B. DeWolf, A. Kemp, R. Werner

Date: 5 August 1983

Subject: Trip Report - Los Angeles

Copies: Distribution

The following trip report is divided into eight sections, each section reporting on each of the meetings we had during the trip. The authors of this report visited the agencies listed below in the Los Angeles area during 13-15 July 1983 in the context of the AIPS technology survey.

(1) Topic: Distributed Operating Systems

At Integrated Computer Architecture (ICA)

With G. Popek

(2) Topic: Software Requirements Engineering

At NASA-JPL

With T. Renfrow, G. Giffen, Y. Yamamoto

(3) Topic: Self Checking Computer Modules

At NASA-JPL

With D. Geer

(4) Topic: Software Engineering

At NASA-JPL

With E. Ng, R. Loesh

(5) Topic: Software Test and Validation

At NASA-JPL

With Tal Brady

(6) Topic: Deep Space Network (DSN) Software Tools

At NASA-JPL

With R. Tausworthe

(7) Topic: Software Fault-Tolerance

At SoHar

With H. Hecht, M. Hecht

(8) Topic: Software Fault-Tolerance

At UCLA

With A. Avizienis

1. Topic: Distributed Operating Systems

At Integrated Computer Architecture (ICA)

With G. Popek

1.1 Background/History

In a distributed systems project at UCLA, a group led by Dr. Gerald Popek has developed, over the past four years, a prototype network operating system based on Unix called LOCUS. At present a 17 node (VAX's) Ethernet network at UCLA is operating.

In the last year, a private corporation, ICA, was founded by Popek to produce a commercial version of LOCUS. ICA supports the UCLA prototype LOCUS in return for a good relationship with the University and continued research.

1.2 Technology Description

LOCUS is a network operating system which makes the network transparent to the user (i.e., it appears to be a single machine). The following features have been incorporated into its architecture.

(1) There is a high degree of user transparency,

(2) It uses Unix 4.1, 4.2, and 5,

(3) It has a full scale DBMS (Berkeley INGRESS),

(4) Transactions (and nested transactions) are handled in the basic operating system, not in the DBMS.

(5) Availability is the key to partitioned operation (e.g., a replicated file can be updated even if all copies are not available).

(6) It is written in C, using a type-checker.

There are, however, some important features not yet implemented.

(1) Processes (and their files) cannot be checkpointed,

(2) Multiprocessors cannot execute in lockstep,

(3) Load leveling is not automatic, presently user controlled.

The LOCUS operating system on the UCLA VAX uses 170K of memory. Unix alone uses 115K.

1.3 Relevance to AIPS

No immediate applicability since their β-test release is scheduled for January 84. We should, however, take advantage of ICA's expertise in distributed operating systems when the AIPS distributed operating system design is developed.

1.4 User Experience

The experience that ICA has had should be useful. The following advice was offered.

(1) Network software is notoriously difficult to deal with. It is important to make the network transparent to the user.

(2) Testing this kind of software is "horrendous". Instrument the code with as many hooks as is feasible to facilitate testing (e.g., checkpoint capability).

(3) Avoid ISO standard multi-layered protocols; instead use specialized problem-oriented protocols.

(4) Bury the network in a device driver to reduce overhead and improve user response; this way it becomes a lightweight kernel process in the OS.

1.5 Status

A commercial version of LOCUS will not be available for AIPS, but the technology for developing a reliable network OS is.

1.6 Future Prospects

Gerry Popek and his associates at ICA and UCLA are willing to give us help if we need it.

1.7 Conclusions/Recommendations

The advice offered (see Section 1.4) by this group is valuable. Since this effort is one of the few successful distributed operating systems available, we should take advantage of this expertise.

2. Topic: Software Requirements Engineering

At NASA-JPL

With T. Renfrow, G. Giffen, Y. Yamamoto

2.1 Background/History

This group of people are part of a systems engineering section at JPL whose function is to provide application projects with support in requirements generation. A major activity of this group presently underway is the generation of requirements for the existing World Wide Military Command and Communications System (WWMCCS).

2.2 Technology Description

Requirements writing for software has traditionally been difficult and error-prone in the software life cycle. Recognizing this, JPL has been using PSL/PSA, an automated requirement tool, as a means of improving this software activity.

The Problem Statement Language/Problem Statement Analyzer (PSL/PSA) was developed at the University of Michigan. The language is used to formalize the system requirements, while the analyzer provides rudimentary facilities for predicting the amount of work needed to solve a problem. The system produces a variety of reports and diagrams useful for program reviews, consistency checking, document generation, and test and validation.

This group at JPL has improved this tool by adding a "friendly" user interface, and some additional reports. They consider this tool a first-generation requirements tool and are engaged, at a low level, in research to develop a second-generation requirements tool.

2.3 Relevance to AIPS

Automated tools in this category (i.e., requirements specification and analysis) are scarce. Both PSL/PSA and CSDL's DARTS are examples of tools which can aid in defining requirements and generating documents yet their use to date can still be classified as experimental. Either tool can be used for AIPS, but it is likely that some customization would have to be done. Appropriate use of tools like these could aid in traceability, testing, and V&V.

2.4 User Experience

JPL has had some success using PSL/PSA on real application programs. They have, however, encountered a problem in training people to develop requirements using PSL/PSA and interpreting the outputs.

2.5 Status

PSL/PSA is available through NASA's software clearing house, the Computer Software Management and Information Center (COSMIC). It does not come with the customized friendly user interface that JPL is using.

2.6 Future Prospects

If PSL/PSA is used for AIPS, the group at JPL would be a good resource for help in customizing the package.

2.7 Conclusions/Recommendations

It is obvious that some automated means of capturing requirements data is desirable for AIPS. Both PSL/PSA and DARTS are candidates.

3. Topic: Self Checking Computer Modules

At NASA-JPL

With D. Geer

3.1 Background/History

On July 13 we visited with Dwight Geer of JPL to discuss JPL's building block approach to fault-tolerant computing.

3.2 Technology Description

JPL's fault-tolerant computer is fabricated by combining appropriate quantities of four building block components into their Self-Checking Computer Module (SCCM). Multiple SCCM modules are connected via redundant system buses to create a distributed computer architecture. The four basic elements are a core, a memory interface, a bus interface and a I/O interface.

Each CORE unit contains two microprocessors run in bit synchronism by a single clock. Instructions and data are fetched from an internal bus. One of the processors is the master and only its data is placed on the internal bus. The other is used to verify operation of the master. Data is compared within the CORE logic and if a disagreement is detected, an interrupt is generated. The fault counter is incremented and both microprocessors restarted. The system uses a software rollback technique and after restart will go to the last established rollback point and re-establish processing. All logic within the SCCM is what JPL calls "Morphic Logic" which causes errors within the logic to be detected as a single error signal at the output. Watchdog processing is employed to detect time outs and invalid instruction execution.

The memory interface building block (MIBB) contains the logic for interfacing the memory storage array to the internal bus. All addresses and data on the internal bus contain a parity bit with the MIBB generating and checking parity for all memory references. Internal to the MIBB a six bit Hamming Code is generated for all words being stored. Upon retrieving a word from memory the MIBB uses the stored Hamming Code to detect and correct single bit errors. When an error is detected, an interrupt is generated to the CORE which uses its software to decide on the appropriate action. There are two spare bit planes within the memory array which the CORE can substitute for failed planes.

The Input Output Building Block (IOBB) interfaces external devices to the internal bus.

The Bus Interface Building Block (BIBB) interfaces the internal bus to the system bus. There is one BIBB for each system bus and it can be controlled externally by another SCCM to gain access to the memory of a SCCM with a failed CORE.

All building blocks have redundant circuitry for the detection of faults. In addition to the CORE fifteen additional blocks (MIBB, IOBB or BIBB) can be connected to the internal bus. All building blocks are tri-stateable to allow them to be spared or removed from the system. The possibility of adding a spare processor to the CORE is being studied.

3.3 Relevance to AIPS

JPL's approach to fault-tolerance is to have hardware detection with software correction whereas the current CSDL approach uses hardware detection and correction. The building block technique, in general, is something we should be tracking because of its ability to be reconfigured and distribute processing.

3.4 User Experience

JPL has had considerable experience in building space satellites.

3.5 Status

JPL has breadboarded a CORE and a MIBB. The CORE is operational and checkout is continuing on the MIBB. A BIBB is planned for FY84. At present another group, Autonomous Repair and Maintenance Subsystem (ARMS), is looking into building a MIBB using gate arrays.

3.6 Future Prospects

JPL's experience in achieving fault tolerance through the building block approach, should be followed for its potential application to some of the AIPS requirements.

3.7 Conclusions/Recommendations

JPL with its expertise in deep space missions is facing many of the problems confronting AIPS. We should continue to maintain contact with JPL and evaluate their techniques for solving the problems of long mission time fault tolerance.

4. Topic: Software Engineering

At NASA-JPL

With E. Ng., R. Loesh

4.1 Background/History

Both of these people work in the Software Engineering and Technology section and deal primarily with Ground Data Support at JPL. They are also responsible for the Galileo Software Testing. No basic research is done in this section and one half of their work is in Ada, for both the Air Force and Army.

4.2 Technology Description

Ed Ng began the briefing by giving us a tutorial on the status of Ada and its support environments. He noted that despite the lagging DoD efforts, commercial Ada efforts (e.g., Rolm, Western Digital) were keeping near to schedule.

He offered the following advice on constructing an integrated life-cycle methodology.

(1) Be very aware of how the different phases interact.

(2) Make sure that tools used in different phases can co-exist.

Bob Loesh discussed a variety of issues including testing, PSL/PSA, and Ada PDL's. Bob teaches a course on Building/Testing Embedded Systems.

Their experience in testing the distributed software for the Galileo space probe has been that it is a difficult task. Testing distributed, embedded systems is a very significant activity, yet there are no mature tools or methodologies to help (they have already spent $2 1/2 million and 4 years on Galileo). The following advice on testing was offered.

(1) Instrument both the hardware and software to make testing possible and easy.

(2) Design for testability.

He referred us to Tal Brady (see next section) for more "war stories" on testing from the Galileo project.

Bob later discussed his experienced using PSL/PSA, a tool which they experimented with.

(1) Learning time is excessive for developers.

(2) It is expensive (if compared to manually produced documentation)

(3) Output is suitable for requirements review but not for a design specification.

(4) Learning time for readers of the documentation is excessive.

He suggested that it is crucial to be able to simulate high level requirements before the design is committed to code. This is not a capability of PSL/PSA, but DARTS does this.

The remainder of the discussion focused on Ada. In his experience, training people to use Ada required the availability of a resident expert consultant for people to use while learning this complex language. He does not like using Ada as a PDL because it forces the software designer to consider low-level issues too early (e.g., typing, range checking).

4.3 Relevance to AIPS

No direct technology input except for the advice of these seasoned software developers.

4.4 User Experience

N/A

4.5 Status

N/A

4.6 Future Prospects

It might be worthwhile for someone to attend the course on Building/Testing Embedded Systems given by Bob Loesh, if it is to be presented in the area.

4.7 Conclusions/Recommendations

We should consider the advice and information obtained from these experts.

Topic: Software Test and Validation

At NASA-JPL

With Tal Brady

Background/History

JPL's Galileo system is a multiprocessor, distributed operating system, deep space probe. It is currently being tested at the integrated level.

Technology Description

The Galileo system is composed of 6 RCA 1802 processors configured in 2 redundant strings. The processors are distributed among HLM (High Level Modules and LLM (Low Level Modules) each of which has its own operating system. It was intended to be verified in parts using a high speed simulator, however, the simulator was never built.

Testing commenced with the first marriage of software and hardware into an integrated system. No provision had been included in its design for facilitating testing (i.e., no clock control and no special "hooks").

It requires all parts of the system to be up and running in order to do any testing at all. Fortuitously, telemetry accesses one common memory address each frame, thus allowing a means of test control and data sync.

Relevance to AIPS

Don't do V&V this way.

User Experience

Mostly frustration.

Status

Behind schedule.

Future Prospects

N/A

Conclusions/Recommendations

(See Relevance to AIPS)

Topic: Deep Space Network (DSN) Software Tools

At NASA-JPL

With R. Tausworthe

Background/History

Bob Tausworthe is the Software Chief Engineer of the Deep Space Network Project at JPL, and also author of Standardized Development of Computer Software, Methods (Vol. 1), Standards (Vol. 2).

Technology Description

The DSN Data System is a geographically distributed network run by the JPL to control satellites and handle data. Each node in the network is a MODCOMP computer running identical software.

Tausworthe was responsible for the institution of the Management and Development Network (MADNET), a homegrown development environment for the MODCOMP which makes software more visible and manageable. This environment uses both commercial and internally developed components. It includes

(1) the Kerrigan and Plauger toolset,

(2) the Caine, Farber, and Gordon PDL,

(3) CRISP, a design and documentation tool (we were given sample output from this tool), and

(4) SOFTCOST, a software cost estimating tool, similar to the COCOMO model.

Configuration management is controlled manually by JPL's Software Management Control Library. We were given a tour of this facility and were impressed by the volume of documentation for this single project.

Relevance to AIPS

Tausworthe and his group are advocates of the design and documentation aid, CRISP. This tool is under consideration, as is its predecessor, NASA's Software Design and Documentation Language (SDDL). Both tools are available through NASA's clearing house, COSMIC.

User Experience

N/A

Status

N/A

Future Prospects

If either CRISP or SDDL is used for AIPS, this group would be a good source of information.

Conclusions/Recommendations

In addition to their tools expertise, this group has potential for AIPS in software cost estimation techniques, configuration management, and software reliability modeling. All of these areas are of concern to AIPS.

Topic: Software Fault-Tolerance

At SoHar

With H. Hecht, M. Hecht

Background/History

Herb and Myron Hecht are the chief advocates in this country for the recovery block approach to fault-tolerant software. Although previously at Aerospace, Herb has now formed a garage-shop operation known as SoHar (for Software and Hardware Reliability).

Technology Description

For each recovery block, there is a primary algorithm, one or more backup algorithms, and an acceptance test. The acceptance test is used to determine when an algorithm has failed. If the primary algorithm fails, a backup algorithm is executed and subjected to the acceptance test. An error return is inevitable if all algorithms fail, but this is presumably a lower probability event than if non-redundant software is used. Hecht characterized this approach as being an orderly way of doing the kind of reasonableness checking that we sometimes do currently. It does not appear that the technique has been used as he proposes on any large-scale systems.

One of the important issues for the recovery block technique is the generation of acceptance tests. In their study of the application of recovery blocks to the FTMP executive, the Hechts used a fault-tree approach to generating acceptance tests. A technical paper summarizing the technique is in the proceedings of the 12th Fault Tolerant Computing Conference.

The primary algorithm, the backup algorithm, and the acceptance test should be as orthogonal as possible to increase error coverage. It should be noted that in Herb's view, the acceptance test may represent an algorithm that approaches the complexity of the primary or backup algorithms.

Herb has defined a Markov reliability model for the recovery block approach. He advocates gathering software reliability data during integration testing by logging errors versus execution time, a technique used by John Musa at Bell Labs.

With respect to hardware support for recovery blocks, he recommends a most-recent-address register at least three deep for memory accesses and something similar for each i/o line. Also, a stack architecture seems best for supporting recovery blocks.

Relevance to AIPS

Recovery blocks are a candidate for structuring AIPS software.

User Experience

There does not seem to be any for real systems.

Status

Available but untested.

Future Prospects

Unknown.

Conclusions/Recommendations

There are obvious tradeoffs between this technique and the n-version technique (see below). In either case, one is attempting to gain greater reliability by expending resources (both developmental and operational). Since three separate versions of the software are required to use this technique, and three would be required as a minimum for fault recovery in n-version, the development resource required would appear to be similar. Operationally, the tradeoff seems to be whether one normally runs only two versions (the primary and the acceptance test) and accepts a longer recovery time, or runs three versions and effectively masks faults in real time. The big question is whether the gain in reliability is worth the additional expenditure of resource. We need to construct some hypothetical situations to clarify the issues.

Topic: Software Fault Tolerance

At UCLA

With A. Avizienis

Background/History

Avizienis is one of the principal advocates of the n-version approach to fault-tolerant software.

Technology Description

In the n-version approach, multiple versions of the software are generated from the same set of requirements. They are run sequentially or in parallel and the results are voted to detect and/or mask faults. Avizienis has conducted student experiments using the technique to investigate specification issues and fault coverage. A rough impression is that 3-version results in something approaching 90% coverage for the errors experienced in relatively untested student programs. The remaining errors are largely due to specification problems.

Avizienis is planning to assemble a team of multinational Post-Doctoral fellows to conduct a dual ethernet experiment utilizing n-version software.

The n-version technique has seen some real-system use in Europe: for a nuclear reactor in Germany, and for the Swedish railway system (DeWolf has references). Reliability models and hardware implications have not really been worked out yet.

His philosophy regarding V&V vs. fault tolerant software is that the n-versions can V&V each other without major tool development and maintenance. An additional reduction in the size of V&V efforts can be accomplished by causing specifications to be so explicit that they can be used for automatic code generation. In this case, a code generator would interpret the specifications and produce "fault-free" code.

This scheme appears to be the amalgamation of at least two concepts, - diversity and redundancy.

Relevance to AIPS

N-version is a candidate for structuring AIPS software.

User Experience

See European experience mentioned above.

Status

Available but many details need to be worked out for running n-versions in parallel. Also specification technology is still primitive.

Future Prospects

Unknown.

Conclusions/Recommendations

See the conclusions under the previous item (Section 7.7).

(Text of CSDL Internal Memorandum)

To: R. O'Donnell Memo No: AIPS-83-58

From: M. Whalen, R. Racine, P. Szulewski

Date: August 2, 1983

Subject: Visit to Softech to Talk About Ada

Copies: AIPS Distribution

TOPIC:

Ada

REFERENCE: AIPS-83-43, "Call from Softech about Ada"

BACKGROUND/HISTORY

The Department of Defense (DoD) has designated Ada as the DoD's standard language for embedded systems and is dedicating a large amount of resources to its development. In an attempt to reduce the costs associated with the development, testing, and maintenance of the software for embedded systems, the DoD has committed an enormous effort to the design of the Ada language specification and the Ada Program Support Environment (APSE).

Softech has a contract with the Army to develop an Ada environment for the VAX system.

As discussed in the referenced memo, a group of Draper people, Paul Szulewski, Roger Racine and I went to Softech on July 26, 1983. Jim Larue of Softech had set up a meeting with himself representing the marketing group, Dr. Larry Weissman representing the technical group responsible for Ada development and Bev Sidler, who is responsible for Ada education at Softech.

TECHNOLOGY DESCRIPTION

Ada is a High Order Language designed and developed for the DoD for use in embedded systems.

RELEVANCE TO AIPS

A requirement of the AIPS program is that the software for the program be done in a High Order Language (HOL). The general nature of the AIPS program, the fact that we are looking to future technologies and the possible application to DoD programs , makes Ada a candidate to be that HOL.

STATUS

Softech has a contract to build Ada systems on the VAX host with two targets: the VAX, and an unspecified computer with a NEBULA architecture. They have written their compiler in Ada, under the VMS operating system. An interim compiler is scheduled to be delivered to the Army in October of this year with the production compiler to be delivered in July, 1984. The interim compiler may be available to CSDL through a couple of sources, that is, the Army or Softech itself.

According to the people that we spoke to on the 26th, they are on schedule and will meet those deliveries. The October compiler is expected to be slow (200 - 400 lines/min.). The July 84 compiler is supposed to be the final production compiler.

The Softech environment was delivered to the Army in September, 1982.

Softech has a contract with Sonicraft to build an Ada with an INTEL 8086 target, scheduled for delivery in September, 1984.

They hope to get a contract to retarget to the YUK44 for the Navy.

CONCLUSIONS/RECOMMENDATIONS

The Ada question must remain open for the present time. Softech, as are all the Ada vendors, is optimistic and enthusiastic about meeting their schedules for delivery of compilers and environments. It remains to be seen if they can.

They have developed extensive training programs ranging from one day seminars for top corporate executives to in depth, hands on programs for engineers and programmers.

They are very willing to help us with training problems. We have people looking at their training programs as well as others.

(Text of CSDL Internal Memorandem)

To: R. O'Donnell Memo No: AIPS-83-61

From: P. Szulewski

Date: 9 August 1983

Subject: Avionics Software Development Environments at AFWAL

Copies: AIPS Distribution, R. Racine, L. Drane, N. Sodano

As part of the AIPS technology survey and working group on software development workstations, I telephoned a group at the U.S. Air Force (AFWAL), Wright Patterson AFB in Dayton, Ohio. This group is responsible to Major Izzy Caro (513-255-3826). The key people I spoke to were

(1) Ray Szymanski - Integrated Support Software System, and

(2) Dan Ehrenfried - Ada Run time Environment for distributed 1750A Architectures.

They can be reached at 513-255-3826.

Ray Szymanski is project leader for the Integrated Support Software System (ISSS). ISSS is a Vax based development system developed by General Dynamics (GD) for them. It is intended to support the development of JOVIAL avionics applications programs. It presently includes: the Interactive Systems (IS) IS/1 workbench, a Unix development system; the Air Force set of JOVIAL tools; and some custom JOVIAL program construction tools. Future (within 6-18 months) tools to be delivered to them through GD include: NASA's SDDL; HOS's USEIT; and a verification tool developed at Georgia Tech. Support for languages other than JOVIAL are also being considered, but not in the near term.

Dan Ehrenfried is they key Ada person and is presently responsible for defining and developing an Ada environment for avionics applications programs. This system is not initially intended for the ISSS, but rather a standalone VAX hosted development station for distributed targets (1750A architectures connected by a 1553 bus). There are three primary objectives guiding this project.

(1) The development of an Ada compiler for avionics applications. This compiler is likely to be a modified version of the Westinghouse VHSIC compiler. They expect a basic capability compiler to be available 15 October, with a final delivery of March 84. Modifications to this compiler will probably be done at Wright Patterson.

(2) Development of an Ada run-time environment to support distributed systems. This includes the executive and other libraries to support real-time, embedded applications.

(3) A methodology for developing distributed software for avionics applications in Ada. They feel that there are few Ada-specific methods and tool available to support Ada program development which take into account the power and features of Ada.

Dan is presently preparing a paper entitled "Incompatibilities Between Ada and the 1750A Instruction Set". He will send me a copy when it is complete.

We originally intended to visit these people to see the ISSS demonstrated. This telecon has given us as much information as is necessary for now. Both of the programs outlined above have little to show now since they are in various stages of planning and procurement.

(Text of CSDL Internal Memorandum)

To: AIPS Distribution Memo No: AIPS-83-81

From: R. Werner, B. DeWolf, and J. Lala

Date: October 26th, 1983

Subject: Visit to BMD Testbed, August 23, 1983

Copies:

Topic

BMD Testbed

Background/History

The Army's Ballistic Missile Defense Advanced Technology Center (BMDATC) contracted with the System Development Corporation (SDC) in Huntsville to set up and run a testbed to experiment with distributed computer architectures for solving the highly time-constrained BMD problem. Two other contractors are involved: TRW is providing a design methodology and design-aid tools, and General Research Corporation (GRC) participates in the area of test and verification. In the morning, Bob Werner, Bart DeWolf, Jay Lala, and Ken Goodwin from CSDL visited Bill McDonald, Howard Welsh, and Cueto Bryant at SDC and toured the facility. In the afternoon, we visited with Henry Minshew and Glen Cox at GRC. The best current description of the testbed is contained in Reference 1.

Technology Description

The testbed hardware consists of a network of 8 Vax 11/780's, a CDC 6400/7600 complex, and a crossbar multi-microcomputer system. The software includes an architecture description language, several operating system components, support software for application programming (Reference 2), and network services. The multi-microcomputer system is composed of 6 Z8001 processing elements interconnected with 12 shared memory buffers by a crossbar switch.

Experimentation with the testbed has been limited to date (References 3 and 4). The plan is to provide support to various BMD contractors in carrying out experiments of interest to BMD. Zary Segall from CMU has been under contract to improve network instrumentation.

Of particular interest to us, SDC has been experimenting with software recovery blocks, although the experiment is being carried out in the context of a simulation with artificial software fault injection. They are interested in fault-tolerant software because the software reliability of the BMD tracking functions in particular is believed to be a major con-

tributor to system performance (i.e., preventing leakage). Thus, backup software in a few key functions can have a large impact on system effectiveness measures.

The following items were of interest with respect to test and evaluation:

(1) The use of TI's Process Development Language (Reference 5), a Pascal derivative, facilitates testing by virtue of its diagnostic capabilities, provisions for test hooks, and runtime traceback support.

(2) The processors in the the crossbar multi-microcomputer system contain register stacks to save the last few instructions.

(3) A real-time clock is available from shared memory.

Relevance to AIPS

The facility may be of interest for point design evaluation or simulation for AIPS. Should this prove desirable at some future time, our use of the facility could probably be arranged. The operating system designs may also be of some interest.

User Experience

Some user experiments have been conducted on the facility, as mentioned above. The color graphics equipment supporting the statistical analysis of these two experiments was impressive.

Status

The testbed is operational, although not at the level of sophistication originally desired.

Future Prospects

The usage of the testbed will probably increase as upgrades to the equipment take place.

Conclusions/Recommendations

We should keep this facility in mind as the AIPS design takes shape.

References (copies available from B. DeWolf)

1. McDonald, W.C., and Smith R.W., "A Flexible Distributed Testbed for Real-Time Applications," Computer, Vol. 15, No. 10, Oct. 1982, pp. 25-39.

2. Vosbury, N. "Design Builder (DB) and Configuration Manager (CONMAN)," TM-HU-288/000/02, Mar. 1982, System Development Corporation, Huntsville, AL.

3. Welch, H. and Moquin, B., "Experiment Results - Data Locking Techniques," TM-HU-311/201/00, June 1983, System Development Corporation, Huntsville, AL.

4. Welch, H. "Experiment Plan - Recovery Block," TM-HU-311/100/00, Sept. 1982, System Development Corporation, Huntsville, AL.

5. Vosbury, N.; and Dingeldine, J., "PDL User's Manual," TM-HU-306/000/02, June 1982, System Development Corporation, Huntsville, AL.

(Text of CSDL Internal Memorandum)

To: R. O'Donnell Memo No: AIPS-83-82

From: B. DeWold and J. Lala

Date: October 5th, 1983

Subject: Visit to IBM, September 15th, 1983

Copies: AIPS Distribution

TOPICS:

Advanced Processor Architecture, Design Verification, Compilers, Operating Systems, and Fault-Tolerant Software

BACKGROUND/HISTORY

The AIPS kick-off meeting in April 1983 which was attended by the Peer Review Group resulted in a letter from Bill Carter to Duncan McIver outlining three areas in which IBM was doing work relevant to the AIPS program. The three areas were as follows.

(1) Specific Fault-Tolerance Techniques,

(2) Architectural Goals, and

(3) Supervisor Program Techniques

In the morning, Bart DeWolf and Jay Lala visited the T.J.Watson Research Center of IBM at Yorktown Heights, N.Y. and had discussions with Bill Carter, Marc Auslander, and Marty Hopkins. In the afternoon we visited their Poughkeepsie facility and talked with Messrs. Bossen, Bahnsen, Carter, Hsiao and Tendolkar.

TECHNOLOGY DESCRIPTION

The IBM researchers at Yorktown Heights have designed and fabricated a 32 bit minicomputer with an internal architecture that is considerably different from those employed by most processors in the past 20 years. The basic thesis of this minicomputer, called the 801, is that each instruction should complete in one cycle. This makes the processor extremely fast. However, only simple instructions such as Register Add, Shift, etc., can be implemented in hardware. The compiler is therefore forced to translate high level language programs into very simple instructions. According to Marc Auslander this actually makes it easy to write the compiler. At the same time the number of

assembly language instructions for a given program for the 801 is not much more than for the MC68000 or IBM 370 machine.

In the area of design verification, IBM machines such as the mainframe CPU 3081 was proved mathematically to produce exactly the functions specified by the automated logic diagrams before even the prototype was fabricated. All 500,000 circuits in the 3081 were exhaustively simulated for this purpose.

In the area of software fault tolerance, we discussed the fault detection and recovery mechanisms in the MVS operating system. These are provided through Recovery Management Support routines which attempt reconfiguration activities in the presence of hardware faults. Functional Recovery Routines (FRR's) provide software fault tolerance via a recovery block scheme.

RELEVANCE TO AIPS

The 801 Instruction Set Architecture is an interesting departure from the standard microprocessor architectures. However, the AIPS program probably does not have the flexibility to invent a new ISA.

Design verification tools of some sort should definitely be used by the AIPS program since this is one of the most powerful methods of culling out generic design faults from massively replicated hardware.

The experience with recovery blocks in MVS would be of use in planning a fault-tolerant software experiment.

USER EXPERIENCE

The 801 processor has been used in research projects to show that it is possible to produce efficient code for this machine and increase the processor throughput considerably over conventional processor architectures.

The design verification tools have been used extensivly by IBM. MVS is of course very widely used.

STATUS

The 801 processor is a prototype and not available commercially. The design verification tools are IBM proprietary and are unavailable other than their high high level description in IBM R&D journal.

CONCLUSIONS/RECOMMENDATIONS

Should we have the freedom to do ISA design in AIPS, the single cycle instruction architecture looks promising from a speed and reliability viewpoint due to less complex hardware.
As far as design verification tools are concerned we should try to obtain these from some commercial vendor since IBM tools are proprietary. In designing a fault-tolerant software experiment for AIPS, we should review experience with the MVS Fault Recovery Routines.

REFERENCES

(1) Radin, G.,"The 801 Minicomputer", Proceedings of the SIGPLAN[9] '82
Symposium on Compiler Construction, Boston, MA., June 23-25, 1982.

[9] Special Interest Group for Languages, of the Association for Computing Machinery (ACM).

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-001

From: R. Racine

Date: November 9th, 1983

Subject: Learning Ada

Copies: Distribution

The new programming language, Ada, seems to be vying with PL/I for the title "Most Complex." This means that Ada will be useful in many different applications, but it also means that it will take users a significant amount of time to learn to use it. I am currently taking a course at ACSI, lasting twelve hours. At the end of this course (we have three hours left), we, I believe, will have covered, in depth, half the language. The fact is, it will be necessary for many CSDL personnel to spend a large amount of time to obtain a working knowledge of the language.

The question then becomes, how does one teach that many people? I have been looking at the different approaches available, and there are many.

The instructor of my course learned Ada at a local college. College courses take about sixteen weeks, for at least two hours per week, generally during the day. It would cost the lab up to $800 per person for this type of course. I have only heard of one Boston area college having this type of course. I do not remember the name of the college.

There is the in-house lecture method, which is approximately what I am taking. It costs the lab the time of the instructor to develop the course and the cost of using a computer. However, it requires a **good** instructor.

There is the outside seminar method, where people go to a hotel for two, three or four days of lectures and, perhaps, practice. Telesoft, Intermetrics, ACM and others are sponsoring seminars, with the cost ranging from about $400 per person, for a Data Processing Management Association two day seminar, to about $850 for an Intermetrics four day seminar. Telesoft offers a four day program, with one day devoted to lecture, and three days of hands-on training, costing about $850 per person. It is possible to bring some of these seminars to CSDL, including the Telesoft course.

The only other formal method of which I am aware is the video-cassette course. I know of only one of these, available from Colorado State University. This consists of ten lectures, lasting thirty minutes each. There is a workbook available, containing notes, examples and problems (with solutions). With a competent person available to answer questions, this could be an alternative to an internal lecture course. Currently, this course can be rented for $1380. The purchase price is $4800. An

C-2

individual tape can be rented for $150 (for a preview). The study guide sells for $22.25 each ($20 each for more than five).

All of these alternatives have good and bad points. The college course is generally thorough and professional. The educational program already exists at the lab to reimburse students. However, college courses are expensive and long. Ada is so new that it is difficult to find the courses.

The internal course is difficult to put a price on, since it would have to be developed, but it can be tailored to the needs of the lab, and can be done with minimal impact to the employees' schedule. The main problem with it is the strain on the instructor to **really** know the language and to be able to develop a good course. This is difficult when there is no working compiler available.

The seminar approach uses experts. They should know the language. The course is also over quickly. However, four straight days of lectures could get rather tiring. Telesoft's course seems to use the best approach, although it would be a good idea to have someone go to it before making a decision.

The video cassette lectures are reasonably inexpensive at $4800, if it is good. If only ten people take the course, the cost is down to about $500 per person (with workbook). It would probably be a good idea to rent a particularly difficult lecture (such as the one on Generic Subprograms and Packages) for $150 to see how good the instructor is.

In summary, these alternatives, with data, are tabularized.

Alternative	Cost	Comments
College	$800 / person	Thorough; Reimbursement program exists; Takes much time; Few courses available.
Internal lecture	Unknown cost to develop course	Can be tailored to CSDL; Minimal impact; Needs knowlegeable instructor, willing and able to take the time to make a good course.
Outside seminar	$400 - $800 / person	Expert instructor; Over quickly; Tiring schedule; Need to find the best one.
Video cassette	$4800 once, $20 / person	Expert instructor; Can tailor to CSDL with help of competent person to give problems; Need to find out if instructor is good.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-002

From: R. Racine

Date: November 9th, 1983

Subject: AdaTEC Meeting

Copies: Distribution

On Thursday, July 7, I attended a meeting of the ACM Greater Boston Chapter AdaTEC. The speaker was Thomas A. Standish, a professor at the University of California at Irvine. The talk was on the use of an Ada Environment developed by a small group of professors and students. It was his belief that much increase of productivity could be gained by using:

(1) An Ada Program Design Language (PDL).

(2) An interactive Ada interpreter for debugging.

(3) A sophisticated database of programs available for re-use.

(4) A program analyzer for timing studies.

The Ada PDL allows the programmer to refine the program from a fairly general "flowchart," with nearly all statements non-executable, to a full Ada program. The methods used to refine the program include:

- Templates for standard constructs (loops, if statements, standard package insertions, etc.). By typing enough characters to distinguish the given template, and then asking for an insertion, the template is immediately retrieved and inserted.

- A method whereby the programmer enters enough characters to distinguish a given variable, and the rest of the variable is automatically generated. This might make long, descriptive names palatable to people who do not like to type. If the variable is not uniquely defined, the common letters will be added and a choice of endings will be given.

The interactive Ada interpreter is somewhat similar to the PLI Checkout compiler. Lines can be added, data can be output, breakpoints can be used. A list of previous statements can be output at breakpoints, or simply the calling sequence (who called whom), to determine how the program arrived at the breakpoint.

Professor Standish had been to Japan recently, and had been quite impressed with their technique of re-using software. They have very large databases of programs. By using keywords, the programmer can find a "similar" program to modify for his purposes. He stated that the Japanese have a conservative metric for productivity, taking into account all the

overhead associated with programming, including the janitors; whereas the U.S. metric is much simpler, taking only a given programmer on a given task into account. It would seem that their programmers have at least three times the productivity of U.S. programmers, even without compensating for the different standards of measurement. He used this statistic to push for databases, and as another reason for having Ada as a standard language. He had not made a database as part of his Ada environment, but was suggesting that it should be created by corporations or the government if the U.S. wants to be competitive.

The program analyzer gathers statistics from a run and creates a bar chart of time spent in each module. It can also color the lines on the program listing to represent how much time was spent on each line. It can, for those without color terminals, give numbers next to each line, instead of coloring the lines.

The environment is not complete; it does not work with tasks, for instance. This is because their compiler is not complete. He does have rather interesting plans for tasks, but nothing written. They used a Digital Equipment Corporation (DEC) VAX computer with the UNIX operating system. I do not know if this configuration is necessary to run this environment. It is possible to obtain these tools. He mentioned sending a tape to one company. Cost was not mentioned.

I was impressed with the tools that he and the others in his department have made. My only question about using this approach at CSDL is whether this type of interactive programming will slow down the MVS operating system too much. It may be more useful on the VAXs or the IBM 4341 with VM/CMS, which are much more efficient for interactive work.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-003

From: R. Racine

Date: November 9th, 1983

Subject: Trip Report of Ada Education at Raytheon

Copies: Distribution

On Thursday, July 13, I traveled with Betty Trakimas and Sid David to Raytheon in Portsmouth, R.I. to talk with Mark Gerhardt. Mr. Gerhardt is in charge of educating Raytheon personnel in Ada. We spent about four hours learning about:

(1) The history of Raytheon's experience with Ada.

(2) Ada as a Program Design Language (PDL).

(3) Ada programming methodology.

(4) His approach to teaching Ada.

Raytheon has been involved with the Ada programming language since about 1979. At that time Mr. Gerhardt asked Professor Peter Wegner of Brown University to give some 1 hour seminars on Ada. The support in the community of people learning about it was so great that these continued until a number of people at Raytheon were more knowlegeable than the professor. They were having discussions on how to do the things currently being worked on if they were to use Ada. That introduced the question of what programming methodology to use for writing Ada programs. They obtained the first videotaped course on Ada, from Honeywell. They suggested changes in this course, which, he believes, are implemented in the new version being created by Computer Thought Corp, in Dallas. He started going to meetings on Ada, and is currently on a number of committees involved with Ada compilers and tools.

He is a leading proponent of using Ada as a PDL. There are a number of Ada-like PDLs being used, including Intermetrics' BYRON.[10] He thinks that Ada is sufficient by itself, and has authored at least one paper on the subject.

We did not have time to go into any detail on methodology, but he indicated that writing efficient Ada programs required using special methods.

---

10 BYRON is a trademark of Intermetrics, Inc.

His thoughts on the available comercial Ada courses are rather negative. For that reason he and four others have developed a series of courses for Raytheon to make certain that they will have the expertise to use Ada when the compilers and programming environments (APSEs) become available. His reasons for not waiting for better courses from outside sources are twofold. First, he believes that they should not wait. He pointed out the DOD Directive 5000.31 (soon to change to 3405.xx), from R. DeLauer, the Under Secretary of Defense for Research and Engineering. This Directive specifies that "Effective 1 january 1984 for programs entering Advanced Development and 1 July 1984 for programs entering Full-Scale Engineering Development, Ada shall be the programming language." Given his feelings that it takes people one full year of working with Ada to become marginally good Ada programmers, he was of the opinion that they had to create their own course immediately.

His second reason for not waiting for a course is his skepticism that anyone will make one good enough for programmers of real-time embedded systems. His experience indicates that none of the teachers of Ada are experts in that area, and thus do not teach it very well.

The courses they have developed are fairly long (over 160 hours, 20 day-long workshops spread over 3 months for an instructor's course). Because of the complexity of the language, he does not think that everyone should try to learn the entire language. But he does think that everyone should learn the Ada methodology. We were able to look at the course curriculum for the instructor's course. He did not seem to be giving too much time for it. He will be presenting a paper at the October meeting of the AdaTEC to give the results of the first course being taught.

He was also able to give us a list of Ada reference books with recommendations. He supplied us with names of people to talk to about video-tapes, PDLs, and general information on Ada.

All in all, it was a highly informative meeting. Mr. Gerhardt is obviously one of the most knowlegeable people in the fields of both Ada and embedded systems. I am looking forward to finding out about his experiences with his course.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-004

From: R. Racine

Date: November 9th, 1983

Subject: Trip Report on Ada-Jovial Users Group Meeting

Copies: Distribution

On Monday to Wednesday, July 18-20, the Ada-Jovial Users Group (Ada-JUG) met in Grand Rapids, MI. Since Monday was to be spent on a tutorial about Jovial, I traveled on that day and went only to the Tuesday and Wednesday sessions. The agenda included an introduction to AdaJUG (starting at 8:15 a.m. on Tuesday), general interest presentations during the day on Tuesday, committee meetings Tuesday night (we kept working until 12:30 a.m.), and committee reports and resolution voting on Wednesday morning. The 1750A Users Group held their meeting starting at 2:00 p.m. on Wednesday, at which time I caught a plane back.

The Ada-Jovial Users Group consists of personnel from: DoD contractors, Jovial and Ada compiler writer companies and those companies involved in making tools and courses for the languages. A majority of the time was spent on Ada issues, With the vast majority spent on the DoD Directive 5000.31, soon to be renumbered 3405.xx (DoDD 3405.xx).

The memorandum dated 10 June, 1983 from R. DeLauer, the Under Secretary of Defense for Research and Engineering, made Ada the language to be used for all projects entering Advanced Development after 1 January, 1984 and those entering Full-Scale Engineering Development after 1 July, 1984. The Ada Joint Program Office (AJPO) sent three people to talk about this Directive, including the Director, R. Mathis; the Navy Deputy Director, B. Schaar; and the Air Force Deputy Director, V. Mall.

Dr. Mathis gave a history of the Directive. It would seem that the General Accounting Office (GAO) and a number of Congressmen believed that the DoD was moving too slowly on the switch to Ada. Pressure was brought to bear on Deputy Secretary DeLauer, who issued the Directive. The Directive is not totally strict; waivers will be allowed.

Col. Mall presented the Air Force view of the impact of this Directive. He suggested two ways of meeting the dates specified in it. The first was the Risk-Management method. Knowing the limitations of availability of Ada tools, a Project Manager can take them into account, adding the cost of developing them or having them developed. If this meant a longer schedule than if a different language were used, so be it.

If the time schedule were critical, a different approach would have to be made. The project could have a duplicate effort, using Ada and some other language. When there is a "clear winner", the other effort could be scrapped.

If the cost of this approach were unacceptable, a waiver would be given. It would not be given lightly, he emphasized.

The Navy had not finished formulating a plan by the time of the Ada-JUG, but LCmdr Schaar suggested that the schedule developed previously to the memo would still be followed. This plan has the first Navy compiler being delivered in July of 1986. When asked if this meant that the Navy was going to give "blanket" waivers to the Directive, the answer was "yes."

Later, during the Ada Liason Committee meeting (nominally from 9:00 to 11:00 p.m.), a resolution was put forth to make certain that Mr. DeLauer found out that the AdaJUG did not think the Directive was in the best interests of the DoD. The exact intent of the membership was debated and voted on (the vast majority supported a positive approach, with reservations as to the lack of a plan, and the lack of taking into account the nonexistence of compilers). Finally, a resolution was written (about 12:30 a.m.), and brought forth to the general meeting the next morning.

This was again debated, and finally rewritten and passed unanimously. The resolution follows:

The Ada-Jovial User's Group strongly endorses the goal of Ada introduction at the earliest possible date. However, for at least a broad class of resource-critical embedded computer applications, the required Ada development tools will not be available in 1984. Therefore, the goals of DoDD 3405.xx cannot be met for these applications.

The Ada-Jovial User's Group calls on the DoD to:

(1) Immediately and significantly increase its commitment to the development of Ada tools, especially those required for resource-critical applications (e.g., airborne systems).

(2) Delay the imposition of DoDD 3405.xx for those areas in which the required tools are not available, and impose it only when such tools have been developed and thoroughly tested.

We firmly believe that failure to take these actions will result in severe cost and schedule impacts, with a consequent loss of credibility for Ada.

There actually were other topics mentioned at the meeting. Softech gave a status report on their Ada Language System (ALS). It is on schedule, with an interim delivery in October, 1983, and a final delivery in July, 1984. The host and target for this system is the VAX 11/780 with the VMS operating system. Shortly after the ALS is delivered, the ALS extension to the Intel 8086 (hosted on the VAX) will be delivered (scheduled for September, 1984). They are also in the process of making an Ada PDL program, doing studies on Ada methodology, making Ada translators to/from various languages, and creating courses to teach Ada.

John Pates of Intermetrics gave a status report on the Ada Integrated Environment (AIE), being developed for the Air Force. Due to various problems, the Air Force gave a stop-work order on everything except the compiler for this fiscal year. The new schedule has the compiler hosted on an IBM 370 architecture machine with UTS, an Amdahl UNIX-like operating system, running under IBM's VM operating system. In August, 1984, the full ANSI Ada compiler will be available, along with a linker, a full screen editor, and possibly a debugger. The full AIE is postponed, pending Air Force funding.

The only other Ada compilers were mentioned in passing. A number of companies are working on other things, like editors, debuggers, etc. The need is great for compilers to test these tools, so these companies are making their own. Whether these will be marketed (or marketable) remains to be seen. It was also mentioned that DEC is looking for Beta test sites for its Ada compiler, and that Westinghouse is making a compiler for the Air Force 1750A architecture computer.

A number of presentations were made in the area of Ada tools and courses.

- Honeywell is making tools for its EMACS screen editor for MULTICS.

- Ed Berard of EVB consulting talked about his courses. He has five Ada courses, including: a one day Introduction to Ada, a five day Analysis and Design for Ada Software course, a three day Ada as a PDL course, a five day Ada Programming Workshop, and a five day Advanced Ada Programming Workshop. He made a number of interesting comments about teaching Ada. He suggests that we are soon coming to the point where one needs to be a "real" software engineer to create good software. His courses, except for the Introduction to Ada, are meant to be taught to people with a background or schooling in the engineering discipline, including various design methodologies, calculus at least through differential equations, linear algebra, plus a few I have forgotten. He also mentioned that his courses are intensive. Many people cannot learn Ada in five days. There are twenty people in the courses, with one instructor; it is difficult to give individual attention to anyone.

- Steven Goings of Computer Thought gave a talk about their Computer Aided Instruction. Using a VAX and a $5000 high resolution terminal with a mouse, this system asks questions about the user's background and then begins to teach Ada. As the "student" progresses, the problems get more complex. The response to errors is based on the user's past experience with other languages: i.e., if the user uses a PASCAL construct which is illegal in Ada, or makes a bad assumption about the order of the storage of arrays based on FORTRAN experience, the program recognizes why the person made the error. An informed response is then given, similar to: "That might work in PASCAL, but in Ada it is:" and then the correct syntax.

From the people I talked to, I received the impression that a lot of work is being done on tools, which is not being publicized. The next year

should see a number of compilers, editors, PDLs, and other tools coming on the market.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-005

From: R. Racine

Date: November 9th, 1983

Subject: Report on Talk by Grady Booch on Object Oriented Design

Copies: Distribution

On Wednesday, August 31, Grady Booch gave a talk at Intermetrics on Object Oriented Design (OOD) of software, especially as it relates to Ada. Bob O'Donnell, Mary Whalen, Nancy Sodano and I were the Draper attendees.

Mr. Booch is one of the experts on the Ada language. He writes a column for the AdaTec group of the ACM called "Dear Ada." He gives seminars around the country teaching the fundamentals of both Ada and OOD. He was in the Boston area to give his seminar at GTE, and was generous enough to accept an invitation from John Pates of Intermetrics to talk to them (and invited guests).

Object Oriented Design is a recent addition to the list of software design methodologies being taught. It is fairly simple in concept: design software from an "object's" point of view. The designer has a "problem space." This is the statement of work, the requirements specification. At the initial stage, the specification might be in English. This design methodology would take the nouns in the specification and translate them into objects (procedures, functions, types, variables, packages, etc.) in the computer language.

His book, Software Engineering with Ada, gives a good introduction to the methodology. It is also a good learning guide to Ada.

The advantages to this approach are many:

(1) The variables and types are easily understood.

(2) Changes tend to affect few compilation units.

(3) Inadvertant run-time errors are minimized - the compiler should catch most of them.

The first advantage is a result of the use of real-world names for functions and variables. The second advantage is a result of the use of abstract data types: if a function does not know or care what the data actually looks like, or how a function works, as long as interfaces do not change, the first function does not care if the data changes or the second function changes internally. The third advantage is also a result of the use of abstract types. Attempting to add APPLES and ORANGES will give a compiler error. Only operations defined for given types are allowed.

The examples given by Mr. Booch were not real-time programs. For those examples, his approach worked well. For embedded systems, it might not work efficiently. For distributed systems, the extra work needed to make a run-time package might be expensive. Space and time constraints seem to be being forgotten. He did not address those issues.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-007

From: R. Racine

Date: November 9th,1983

Subject: Presentation of Computer * Thought Ada Products

Copies: Distribution

On Monday, September 12, Steven Goings and Truman Blocker of Computer * Thought came to Draper Lab. to present some of their products scheduled to be coming to market this fall and spring. The products include:

(1) An Ada interpreter for the VAX.

(2) A symbolic debugger for the interpreter.

(3) Ada * Tutor,[11] an automated teacher of Ada.

(4) A videotape course on Ada.

(5) An Ada seminar/workshop.

The presentation took about an hour and a half, and consisted of 35 mm slides, overhead slides and a demonstration on an IBM personal computer (PC). Mr. Goings first gave an overview of Computer * Thought, which is an offshoot of Texas Instruments. He then presented information and slides pertaining to the subjects. Finally, we took him to a Lab IBM PC to allow him to show us graphics aids used during their seminars. He left us with manuals with details of their products, including a price list.

The products were, in my opinion, quite good.

(1) The Ada Interpreter. For learning, an interpreter is an excellent aid. Their interpreter has almost all features of Ada, missing only task types (it has task objects. Task types are extensions of task objects, and would be easy to teach without their use), task priorities, representation specifications (the ability to tell the compiler how to represent variables, such as storage location, bit configuration, etc.), unchecked type conversions and some other constructs not essential to the validation of the interpreter. By Spring of next year, they will probably have a validated product.

The only problem that I could think of in terms of using it as a learning aid is that it is slow. He showed an example of a program about 30 lines long. It took about 60 seconds to compile, and

[11] Ada * Tutor is a registered trademark of Computer * Thought.

another near minute to run. It was not doing a lot of work. He pointed out that the example was being run on a different machine than the final product will use, and estimated that the actual product would be at least twice as fast.

(2) The symbolic debugger. It seemed as good as others I have seen. It allows the user to see the source (although it does not allow modifications to the source while in the debugger). One can set breakpoints on statements or variables. One can look at the value of any variable. One can restart the program at the beginning. One can look at the calling history.

(3) Ada * Tutor. This was the best of their products. Using Artificial Intelligence techniques, this program teaches a user how to use the system and how to program in Ada. The way in which a person is taught Ada is, I think, unique.

The program asks questions about one's experiences with other languages. This starts the building of a database containing knowledge about the student. The student can then try to build a program which solves one of the problems "known" to the system.

The problems are given in a document. A list of prerequisite problems and readings (if any) are given in the document, along with the statement of the problem, an algorithm for solving it, and a "code template" and expected identifiers, so the computer system can access it correctly.

When the student finally creates a solution to one of the exercises, the system can test it. Not only are the outputs checked against the correct outputs, but the LOGIC of the program is checked, by tracing through the program as it is running.

For instance, in a bubble sort routine, which takes an unsorted array and puts the elements in order, not only is the resultant array checked for correctness, but the number of switches of elements in the array is counted. If more than the optimal number are used, the student is told about it. If the system can determine a reason, such as switching if the elements are equal, it will display that information as well.

It is probably necessary to try the system before saying that it is really worth buying, but the impression given was that it is excellent.

(4) The videotape course. Unfortunately, it was recorded in a format not compatible with our players. We therefore were unable to preview the course.

(5) The seminar/workshop. This also seemed to be very good. It consists of five days of in-house lectures and hands-on experience. "The seminar content is designed around a solution to a generalized message switch case study problem." The lecture is augmented by graphics generated on TV screens positioned in the lecture room.

The hands-on experience is gained using their interpreter/debugger. We were shown some of the graphics on an IBM PC.

The products seemed to be well designed. Some questions need to be answered by looking at the finished product, such as how "smart" the Ada * Tutor actually is. Other questions need to be answered internally, such as our method of teaching Ada to those who need to learn it.

It is very likely that the finished product will be present at the AdaTEC/AdaJUG meeting in October in Dallas. We should be able to find an answer to the first set of questions there.

(Text of CSDL Internal Memorandum)

To: Distribution Memo No: ADA-008

From: John S. McLachlan

Date: November 9th, 1983

Subject: Accessing Telesoft Ada Through SPF

Copies:

This memo describes how to access the Telesoft-ROS[12] via SPF panels.

(1) Setting up your TSO session to access ADABOOT panels

These panels and clists have been developed to speed up and simplify the ROS load process. The panels and clists have been stored under the primary index FTP. To gain access to the ADABOOT panels, the user's CMD.CLIST(ZLOGON) should be edited to include the argument FTP in the XSETUP command in each of the three parameters; panel, clist and msg. For example, if the user's zlogon includes the following XSETUP command:

```
XSETUP CLIST(userid)
```

it should be changed to

```
XSETUP CLIST('userid,FTP') PANEL('FTP') MSG('FTP')
```

where "userid" is the user's TSO id (ie.,JSM1269).

(2) Using the ADABOOT panels

To access the ADABOOT panels enter the 'U' option from the SPF Primary Option Menu. This will call up a menu which should offer a selection code to call the ADA panel. If you do not have a VUSER panel, the FTP VUSER panel will automatically be used and the correct ADABOOT option(s) displayed. If you do have a VUSER panel of your own, the ADABOOT selection should be placed there. The selection command 'N','cmd(adbtin)' will call the ADABOOT panel sequence, where N is the selection number/letter in your own menu.

(3) The ADABOOT panel

Once inside the ADABOOT panel, there are five main input sections:

---

12 Telesoft-ROS is the operating system through which users may compile and execute programs. See also (CSDL) DCD memo #771 "Using Telesoft Ada at CSDL".

- Command line
- Remread dataset
- Profile prefix
- Rosvol files
- Terminal input/output

Terminal input/output

(a) Command line

There are two possible command line selections for ADABOOT. B for ADABOOT/NOREMREAD and R for ADABOOT/REMREAD. Both commands load and execute the ROS, but "R", ADABOOT/REMREAD processes the specified remread dataset such that it may be copied onto one of the ROS volumes via the ROS REMREAD command.[13]

(b) Remread dataset

The Remread option takes the user input dataset and copies it into a new dataset with special ROS compatible attributes.

If the user is not going to be issuing a remread command, he/she can choose the BOOT/NOREMREAD option on the panel, avoiding a dataset check and a copy command (saving time and money).

(c) Profile prefix

This input specifies the prefix that the panel will assume when allocating datasets for the ROS files. When the ADABOOT session is over, the profile prefix is set back to the profile prefix the user was currently working under.

(d) Rosvol files

The ROSVOL inputs specify the names of the datasets allocated as the four volumes ROSVOL1, ROSVOL2, ROSVOL3 and ROSVOL4. The default values are BOOT, MOD, ROSVOL3 and ROSVOL4 respectively, specifying the datasets 'prefix.BOOT.ROSVOL', 'prefix.MOD.ROSVOL', etc.[13]

(e) Terminal I/O

---

[13] See DCD memo#771 "Using Telesoft Ada at CSDL"

ROS communicates with MVS via TERMIN, TERMOUT and PRINT.

The TERMIN allocation is for standard ROS input. The default value is for input from the terminal. If allocated to a dataset, the dataset must exist.

The TERMOUT allocation is for standard ROS output. The default value is for terminal output. If allocated to a dataset, the dataset will be created if it doesn't exist.[14]

The PRINT allocation is for use with the ROS PRINT command. The default for print is SYSOUT, but this can also be changed to another printer (ie: SYSOUT(x)) or to a dataset. If PRINT is allocated to a dataset that doesn't exist, it will be created.[15]

Once the panel selection is made, the datasets are checked to see if they exist. If no errors occur, the ROS load module is called and ROS is booted. If errors should occur, an error panel is called and all errors are displayed on the screen along with a small explanation and an option for a further in-depth explanation and additional help.

---

[14] All existing termin/termout datasets must be blocksize 80.

[15] Note - If the profile prefix is changed, the computer will look for/create the terminal I/O files in the new prefix unless the dataset(s) is/are enclosed in quotes.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-009

From: R. Racine

Date: November 9th, 1983

Subject: Presentation of LabTek Marketed Ada Products

Copies: Distribution

On Thursday, October 6, I attended a demonstration of an Ada workstation marketed by LabTek. Thomas Griest, LabTek's president, gave the demonstration. He first covered a few aspects of the Ada tasking facilities. Then he talked about LabTek, Telesoft (which created the Ada compiler and ROS operating system), and WICAT (which makes the workstation).

LabTek is located in Westport, Connecticut. Their purpose is to market and support Ada workstations for DoD contractors. Because the DoD is mostly interested in Ada for embedded systems, LabTek is emphasizing that application in tools they are building and in the support they give.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-010

From: R. Racine

Date: November 9th, 1983

Subject: Trip Report - AdaTEC/AdaJUG Meetings - Teaching Aspects

Copies: Distribution

During the week of October 17-21, Betty Trakimas, Alton Knosp and I attended the meetings of the Ada/Jovial Users' Group (AdaJUG) and the Ada Technical Committee (AdaTEC) of the ACM. The AdaJUG is a group of personnel from: DoD contractors, Jovial and Ada compiler writer companies and those companies involved in making tools and courses for the languages. The AdaTEC is the Technical Committee for Ada within the Special Interest Group for Languages (SIGPLAN) in the Association for Computing Machinery (ACM).

Anyone is welcome at either group's meetings. The voting membership of AdaJUG consists of all attending. The voting membership of AdaTEC consists of all members of AdaTEC (there is a membership fee).

This is the first of a series of memos on the subjects covered at the meetings. This memo will cover the teaching of Ada.

There were two tutorials on Monday. The first was an introductory tutorial on Ada, by Putnam Texal of SofTech. She gave a very quick overview of the language in a Top-Down approach. She first went over the general design of the compilation units. Then she covered the use of each type of unit. Only after we had covered the design of programs and knew how to use tasks, procedures, packages and generics did she go into the "executable code," and teach the syntax of loops, conditionals and the other lower level constructs.

I was quite impressed with this approach. She was able, from the beginning, to teach abstraction methods, information hiding and structured programming, Software Engineering methodologies which are easy to use with Ada. Her point of view was that it is easy to learn the syntax of a language, but it is more important to learn the ways to write readable, maintainable code.

That afternoon, Grady Booch gave a talk on Ada style. We learned how to name identifiers for readability. He gave us the techniques of Software Engineering made easy by using Ada.[16] He said that, in his experience teaching Ada, the difficult parts of the language are:

---

16 Abstraction, Information Hiding, Modularity, Locality, Completeness, Confirmability and Uniformity.

(1) The complexity of the language.

(2) Typing / data abstraction.

(3) Concurrency.

(4) Packages / generics.

Mr. Booch teaches object-oriented design of software. This approach uses abstraction, information hiding, modularity and locality to make readable, maintainable code. In this method, each module attempts to model an abstract object derived from the problem definition. Packages are "objects", procedures and functions are operations, types are classes or "common nouns" and variables are "specific nouns".

After championing this design scheme, he gave some good advice about the "little things" that make the life of the software developer and maintainer easier.

(1) Indent the code according to some standard (such as the Language Reference Manual (LRM).

(2) Write with the reader in mind, not the writer (it is usually written in toto only once, but read many times).

(3) Do not use numbers when there is the remotest chance they might change value sometime in the future. Use declared constants instead.

(4) Use the Ada Attribute capability whenever possible (Ada allows the programmer to say "type'range" to signify the range of an array, instead of needing to say "1 to 100", for instance).

(5) Use all the typing features of the language. Do not use the predefined types, except in new types.

(6) He gave an interesting device for making compile-time assertions (he advocates finding as many errors as possible during compilation) using Ada's Discriminated Records.

(7) Name subprogram parameters (Value := Pop(Stack => My_stack); This means that the function Pop is called with the formal parameter Stack given the value of My_stack. With many parameters, this ability to name parameters makes errors much less likely).

(8) Use Exceptions for "error conditions".

(9) Use fully qualified names (So you know where things are declared).

There was a panel discussion on style later in the week. Some people hated the underscore (too difficult to type) in identifiers. Some thought comments were virtually unneeded in good Ada programs; others were of the more traditional "lots of comments" school. Some thought GO TO's were never needed in the language; others could think of certain places where it was more readable to use a GO TO than to do anything else. These dif-

ferences were minor. The key concept everyone agreed with was that programmers should write readable code with a consistent style. Identifiers should be descriptive (no R1's, ECNTR's. Use real words like Drag_reference or Energy_counter). The use of GO TO and EXCEPTION should only be done when absolutely needed FOR READABILITY (if at all).

I have already mentioned that Putnam Texal taught Ada in a top-down manner. I heard no arguments against that approach, and one very good argument for it. It came during a talk by Jim Miller of Computer * Thought on Computer Aided Instruction (CAI). Their company is developing Ada * Tutor, a CAI product using Artificial Intelligence (AI) to teach Ada. Mr. Miller has a degree in psychology. He suggested that the way one learns complex subjects is to form a model using known concepts; then learning involves correcting the model. So the important thing is to get the student to have the closest thing to reality as the initial model, or else one might have difficulty changing the model. As an example, if the student knows PASCAL and FORTRAN, it would be much better to start teaching Ada by comparing it with PASCAL than by comparing it with FORTRAN. If the student only knows about FORTRAN, one should start with abstract concepts totally foreign to FORTRAN, such as Packages. By saying that Ada is "like" packages in one's attic, the student is less likely to have the idea that it is in the least bit similar to FORTRAN (or MAC, or HAL, or PL1, or ...).

So once the student has an abstract idea of Ada as a bunch of packages, it is easy to modify the idea by teaching package interfaces, known as "specifications". If, on the other hand, one starts by teaching the basics of what an assignment statement looks like, or where semicolons are required, one is reinforcing the model of Ada as "like" FORTRAN, making the modification of the model more difficult in the end.

On Thursday, I went out to Computer * Thought to see a demonstration of Ada * Tutor. I was impressed with the concept. I was less impressed with the product. My only hope is that they change the product before marketing it (they were quite receptive to constructive criticisms).

The problems, in my opinion, with the product as presented are:

(1) It teaches in a bottom-up manner.

(2) It checks logic, but not style (one could have all one-letter identifiers or use GO TO's and it would not complain).

(3) There are no "windows". It has the ability to look at documents stored in a file. Unfortunately, one can not look at the code at the same time.

(4) It has a "syntax oriented" editor with "templates". One can pick a given template for an Ada construct (If Then Else End if, for example) and the editor will put it in the code at the specified spot. What is more interesting (and unfortunate?) is that it keeps a model of the student, and "knows" what templates the student can use. It displays only those templates, and the student can not use any other templates than are displayed.

(5) It may be slow. They demonstrated it on a Symbolics LISP machine. It is going to be targeted to the VAX. The VAX 11/780 is much slower than the Symbolics machine, according to Jim Miller. He does not know how much slower the VAX version of the Ada * Tutor will be, but it could be one-fifth the speed of the Symbolics version.

Two managers from IBM's Federal Systems Division (FSD) gave a talk on their experiences using Ada as a design language. Since it was their first time using the language, they were very conservative. They gave courses in software technology (abstraction, information hiding and data encapsulation in particular), Ada, IBM's design levels and IBM's programming standards. They had outside experts criticize their initial attempts at programming in Ada. They had internal reviews.

Their conclusion was that it takes about four months to obtain a competent Ada programmer, about one month of formal training and three months of Ada programming.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-011

From: R. Racine

Date: November 9th, 1983

Subject: Trip Report - AdaTEC/AdaJUG Meetings - Implementations

Copies: Distribution

During the week of October 17-21, Betty Trakimas, Alton Knosp and I attended the meetings of the Ada/Jovial Users' Group (AdaJUG) and the Ada Technical Committee (AdaTEC) of the ACM. The AdaJUG is a group of personnel from: DoD contractors, Jovial and Ada compiler writer companies and those companies involved in making tools and courses for the languages. The AdaTEC is the Technical Committee for Ada within the Special Interest Group for Languages (SIGPLAN) in the Association for Computing Machinery (ACM).

Anyone is welcome at either group's meetings. The voting membership of AdaJUG consists of all attending. The voting membership of AdaTEC consists of all members of AdaTEC (there is a membership fee).

This is the second of a series of memos on the subjects covered at the meetings. This memo will cover the implementations of Ada.

Intermetrics and SofTech are still expecting to have compilers ready for validation by mid summer of 1984. By September, Softech should have their full Ada Language System (ALS) delivered to the Army. Intermetrics, however, still is waiting for money to continue their work on the Ada Integrated Environment (AIE) tools, other than the compiler. According to Don Roberts of the Rome Air Development Center (RADC), which has the contract with Intermetrics, they are carefully considering their priorities. It is possible that the creation of a 1750A targeted compiler will be found to be of higher priority than the creation of an environment.

The Air Force Wright Avionics Laboratory (AFWAL) is obtaining bids for the creation of a 1750A targeted compiler. The 1750A instruction set received much attention, with many questions raised about the problems of making a validated compiler for it. One of the most interesting questions was:
For which 1750A processor does one validate?
The 1750A Standard is flexible. It allows the implementer to add features, including new instructions. This question has yet to be answered by the Ada Joint Program Office (AJPO).

Christian Rovsing A/S (a Danish firm) is developing Ada compilers hosted on the VAX, targeted to various processors, including the VAX. They expect to validate in the Spring of 1984.

New York University is working on an upgrade to their AdaEd interpreter, making it much faster for both compilation and execution. Their first priority is to be the first compiler to re-validate in April, 1984. They are planning a number of upgrades after that, including a complete re-write of the compiler in C.

Telesoft still plans to submit a compiler to AJPO in January, 1984 for validation. They had a demonstration of their syntax and semantic checker for complete Ada, on a VAX. It worked, seemed reasonably fast, but it was just a demonstration. The real test will come in January.

Rolm also had a "hospitality suite", containing a Data General computer (I do not remember the model) and about six terminals. The compiler was slow, the editor was not the most easy to use, the error messages were not too helpful. It is a validated compiler, however, and is targeted to DoD qualified processors.

In conclusion, everything is still running on about the same schedule as was given in the July AdaJUG meeting.

(Text of CSDL Internal Memorandum)

To: Computer Purchasing Committee Memo No: ADA-012

From: Alton Knosp

Date: November 10th, 1983

Subject: Trip Report -- AdaJUG/AdaTEC Meetings -- DoD Ada Policies

Copies: Distribution

This memo summarizes the separate policies of the Navy, Army, and Air Force concerning the use of the Ada computer language. The policies will require the use of the Ada language and DoD sponsored Ada Programming Support Environments (APSEs) when developing software for embedded computer systems. The three services differ, however, in timing and, initialially, the extent to which the Ada language and APSE's will be required.

The policies were prompted by a memo from Under Secretary of Defense Robert Delauer changing DoD directive 5000.31 to mandate the use of the Ada language for all mission critical Defense applications starting advanced development after January 1, 1984 or starting full-scale engineering development after July 1, 1984. Representatives from each of the services described the policies at the joint AdaJUG/AdaTec meeting in Dallas, Texas held from October 17 through October 21, 1983. The meeting of over five hundred people was attended by Betty Trakimas, Roger Racine, and myself.

A.1 Air Force Policy

Major Al Kopp outlined the following Air Force policy regarding DoD directive 5000.31:

- The directive mandates the use of Ada for programs of the Defense Systems Acquisition Review Committee (DSARC) and Air Force Systems Acquisition Review Committee (AFSARC) with phase 1 milestones starting after January 1, 1984 and phase 2 milestones starting after July 1, 1984. Mandated programs must:

  (1) have a validated compiler and acceptable support environment before the contract is awarded, or

  (2) provide funding for Ada compiler and environment development that will be completed prior to the Critical Design Review (CDR), or

  (3) provide funding for a duplicate development effort using Ada and an approved higher-order language with Ada to be used when compiler and environment development are completed.

- All non-DSARC/AFSARC programs must comply to the maximum possible extent consistent with validation and the Air Force Four-Phase plan:

  (1) teach Ada to personnel first -- contractors are responsible for training their own people,

  (2) begin using parallel development (Ada and another HOL) for small programs,

  (3) work up to small low-risk projects, and

  (4) finally, require Ada for all embedded system software.

The Air Force already considers that the mandate requires the use of Ada for a number of programs[17].

As a matter of policy, the Air Force will not provide software tools or environments as Government Furnished Equipment (GFE). These may be provided for free to some educational institutions. (In other words, the Air Force will not be responsible for any software products provided for free.)

The above policy was presented by Major Al Kopp, HQ USAF/RDST (202) 694-8250.

A.2 Army Policy

All embedded computer systems for the Army will be required to utilize:

- an Army Ada compiler,
- the Army APSE, the Ada Language System (ALS), currently hosted on the DEC VAX, but with plans pending for rehosting, and
- an Ada oriented program design language (PDL).

The Army plans to provide all of its Ada compilers and the ALS as GFE. The ALS (including the interim Ada compiler) is available now to anyone volunteering to rehost it on another machine or retarget the compiler to another machine.

---

[17] The following project acronyms were listed at the meeting as requiring the use of the Ada language: AMPE, COMBAT ID, enhanced JTIDS system, WWMCCS, WAAM, combat IFF, ATF, HAVE CLEAR, JVX WWMCCS MOD, AMLS, and JSTARS.

The above policies should begin in accordance with directive 5000.31 as outlined in the Delauer memo. The Army compiler for the ALS is scheduled to be validated in the fall of 1984.

The above policy was presented by Mr. H. Archibald. For more information, contact the Ada Joint Program Office (AJPO), Rm. 3D139, (400 A/N) Pentagon, Washington, D.C. 20301, telephone (202) 694-0209.

A.3 Navy Policy

The Navy recognizes that Dr. Delauer's memo has not yet been officially signed as a DoD directive. Therefore, they are currently issuing waivers to allow contractors to use Ada since Ada has not been officially added to the list of approved higher order languages for the DoD.

Assuming that the directive will be signed, the Navy plans to require the following for embedded computer system software and any other software developed for the Navy:

- Ada compilers targeted to Navy family of processors (including the MC68000),
- the Navy version of the Army's APSE, ALS/N, and
- an Ada oriented design language.

The Navy plans to utilize the Army's APSE (the ALS) with special Navy additions to produce ALS/N. Additions include compilers targeted to Navy machines and some software tools that the Army did not include in the original ALS. The Navy also plans to eventually develop standard operating systems for each Navy approved processor. However, Navy funding for Ada has recently been cut from the proposed Federal budget.

The above policy was presented by Mr. D. Boslaugh. For more information, contact the Ada Joint Program Office (AJPO) (address given above).

(Text of CSDL Internal Memorandum)

To: Ada Committee Memo No: ADA-013

From: Alton Knosp

Date: November 10th, 1983

Subject: Trip Report -- AdaJUG/AdaTec Meetings -- Compiler Validation

Copies: Distribution

The DoD policies towards Ada compiler validation may have significant effects not only on the maintenance of Ada compilers but also on the design and modifiability of hardware for which code is generated. This memo lists those policies that were described at the last AdaJUG/AdaTec meeting and the questions that were raised regarding them. The meeting of over five hundred people was held in Dallas, Texas, from October 17 through October 21 and was attended by Betty Trakimas, Roger Racine, and myself.

The use of the Ada language is mandated by the DoD directive 5000.31 for all mission critical Defense applications which begin advanced development after January 1, 1984, or full-scale engineering development after July 1, 1984. Navy, Army, and Air Force policies, based on this directive, will require the use of the Ada programming language, and, for the Army and Navy, an Ada-like Program Design Language (PDL) as well as an Army or Navy approved Ada Programming Support Environment (APSE). There are no current plans to validate a PDL or an APSE, though work is proceeding towards defining an APSE standard.

All the services have agreed to use only Ada compilers that have been validated by the Ada Validation Office (AVO) of the Ada Joint Program Office (AJPO). The AVO policies towards validation were outlined at the conference as follows:

(1) AVO Validation Policies

(a) The Purpose of Validation

The purpose of validation is to eliminate the possibility of proliferation of different dialects of Ada. The AVO's purpose is to provide technical assistance to language implementers and to research areas of software validation and quality assurance.

(b) The Validation Process

To be validated, a compiler must first be certified. A compiler is certified when a suite of tests from the current version of the Ada Compiler Validation Capability (ACVC) is completed successfully. The compiler implementer has ninety days after certification to have the compiler validated.

After certification, a team from the AVO will test the compiler using a different suite of tests from the ACVC. The costs of this validation will be borne by the implementer of the compiler.

For periods of time before, during, and after the execution of the ACVC, the implementer may challenge the tests. The implementer may also challenge the AVO analysis of the results of the ACVC tests during a short grace period following release of the analysis.

The AVO may also waive certain portions of the ACVC which are not applicable to a specific target, such as tests for performance of text I/O for embedded micro-processors.

(c) Required Re-validation

A compiler must be re-validated within a year after its previous validation.

A compiler may be waived from re-validation for up to two years provided both the compiler and the ACVC have not been changed.

If a compiler fails a new set of validation tests, it loses its validation.

If the target machine for the compiler has been modified, the compiler must be validated for the "new" target.

(d) Completeness of the ACVC

The ACVC is being constantly reviewed and revised. Two more updates are scheduled for 1983 and two during 1984.

The Ada Language Reference Manual (MIL-STD-1815) is the definition of the language, and, thus, the ACVC. The manual states that certain features are optional, such as low level I/O, pragmas, and representation specifications. These are not tested by the current ACVC and are not planned to be included in the near future, though the AVO is considering the addition of pragmas.

(e) Validation and Hardware

A validated compiler has one host machine and one target machine (possibly the same). This means that compilers with multiple targets require multiple validations.

If the hardware configuration for an intended target is modified, it is up to the implementer to prove that the change does not affect compiler validation. Otherwise, the configuration is viewed as a second target requiring separate compiler validation at re-validation time.

(2) Questions Raised and Points Made at the Conference

Validation is NOT synonymous with useability. The Language Reference Manual leaves many options up to the whim of the implementer, especially scheduling algorithms for real-time tasking.

A question was raised concerning policy towards compilers which lose their validation just before a project is about to be completed. To the AVO, the project would be utilizing a non-validated compiler. Both the AJPO and the three services mentioned that this area required more study.

A question was also raised concerning how much a target machine configuration would have to change before a given compiler would lose its validation for the reconfigured target. The AJPO and services stated that this area also required further study.

(3) Conclusions

The DoD policies concerning the use and validation of Ada compilers will increase the impact of the compilers on projects requiring software. The design of the system executive, for example, has been moved from the embedded software design into the compiler. Design decisions for hardware may be limited to keep the configuration within the "target" limits of a compiler or within the limits to which a compiler can be modified or purchased.

The above policies will favor the production of compilers which have vendor support (or are customer modifiable) for:

- revalidation due to changes in the ACVC,
- re-targeting and validation due to target hardware re-configuration, and
- re-designing scheduling algorithms and run-time support packages.

(Text of CSDL Internal Memorandum)

To: P. Felleman Memo No: ADA-017

From: R. Racine

Date: November 30th, 1983

Subject: Trip Report - AdaTEC Local Meeting

Copies: Distribution

On November 11, 1983, I attended a meeting of the local section of AdaTEC, the Technical Committee for Ada within the ACM. The featured speaker was George Cherry, one of the noted Ada tutorial instructors.

Before the meeting, a group went to Joyce Chen's for dinner. I was able to obtain information about Raytheon's and Intermetrics' current activities, along with the latest gossip concerning various compilers' good and bad points. Raytheon continues to give courses; Mark Gerhardt is giving a graduate course at the University of Rhode Island. Some of the people he taught this past summer are teaching Ada to the "troops."

Intermetrics still does not have funding for the full AIE; they are still working only on the compiler. It is on schedule and will be available in August of 1984. They are negotiating with other organizations for retargeting jobs.

The gossip concerning the state of various compilers was depressing. SofTech is supposedly having difficulties with efficiency. Telesoft may have difficulty validating, and there was skepticism as to how many of the "extras" would be implemented (like interrupt handling, which is optional). Everyone was skeptical of Westinghouse finishing the 1750A compiler.

The meeting itself was interesting, up to a point. The topic covered by Mr. Cherry was "The Object Flow Method". He tried to show how to take Data Flow Diagrams (renamed Object Flow Diagrams because Ada calls "data" "objects") and turn them into Ada programs, packages and tasks. Unfortunately, a number of people did not like Object Flow Diagrams, and entered into a rather heated and lengthy debate as to whether they were of any use, since they give a rather limited view as to what is happening in the system.

The method used was not all that complex. Most of the time was spent on examples. The Method consists of:

(1) Given the statement of the problem, decompose the processes into a hierarchical object flow diagram. First, decide what processes exist. Then draw boxes with inputs and outputs from each process as arrows between boxes. Decompose each process until each process is "primitive", or obviously should not be broken down any further.

(2) Take the diagrams and create a "main" program, consisting of the outermost constants and data types and containing packages corresponding to the processes in the top level Object Flow Diagram. Within each package will be the procedures and tasks and packages corresponding to lower level diagrams.

(3) List all interfaces. Ada does not require the programmer to specify within a package which procedures raise which exceptions or which procedures call which other procedures. At this point, all these interfaces should be documented with comments.

(4) Finally, write the code for each procedure, task and package.

The major problem with this approach was its decided "Top-Down" aspect. Many people suggested that he never talked about generics, or library packages in general. For instance, he mentioned that a number of his processes needed buffers. He never mentioned that a generic buffer could be written; he implied that a new one would be written for each application.

It was obvious that if one were familiar with and liked Object Flow Diagrams, one would like Mr. Cherry's method of creating Ada programs from them. If one was not familiar with them, like me, it was an interesting introduction. To those who already had an alternate design approach, this seemed a bad approach.

APPENDIX B

LITERATURE LIST

This appendix presents the literature list which resulted from the literature-search portion of the technology survey.

The literature list is presented in two parts. The first part is sorted alphabetically by authors, then by document date for each author of more than one listed document. In this first part, each listing is limited to title, authors and document date.

The second part is sorted alphabetically by title. It contains additional information from the document records stored in the electronic filing system from which these reports were generated. Specifically, each entry contains the following information if available:

Title,
Document Type,
Document Date,
Authors,
Source, Pub. by, etc,
Filed by,
Document Number, and
Abstract

To simplify inclusion of this appendix, its pages after B-2 are not numbered in the conventional manner (B-3, B-4, etc.). The pagination supplied in the reports produced by the electronic filing system (XFILE) is used instead (Page 1, Page 2, etc. for <u>each</u> of the two parts.)

The two parts are easily distinguished by noting that the title of each report is imprinted at the top of each page generated by the XFILE system, and also by observing the difference in format of the entries in the two reports.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

ADAMS, M.
MEASURES OF MERIT FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: 1983
AUTHORS: GAI, E. G. ADAMS, M.

ADRION, W.
VALIDATION, VERIFICATION, AND TESTING OF COMPUTER SOFTWARE
DOC. DATE: JUNE 2, 1982
AUTHORS: ADRION, W. BRANSTAD, M. CHERNIAVSKY, J.

AGGARWAL, K.
RELIABILITY INDICES FOR TOPOLOGICAL DESIGN OF COMPUTER COMMUNICATION NETWORKS
DOC. DATE: DECEMBER 1981
AUTHORS: SOI, I. AGGARWAL, K.

AGRAWAL, D. P.
DESIGN OF SOFTWARE FOR DISTRIBUTED/MULTIPROCESSOR SYSTEMS
DOC. DATE: 1982
AUTHORS: MCKELVEY, T. R. AGRAWAL, D. P.

ALFORD, M. W.
A REQUIREMENTS ENGINEERING METHODOLOGY FOR REAL-TIME PROCESSING REQUIREMENTS
DOC. DATE: 1977
AUTHORS: ALFORD, M. W.

ALLAN, R.
LOCAL NETWORKS: FIBER OPTICS GAINS MOMENTUM
DOC. DATE: JUNE 23, 1983
AUTHORS: ALLAN, R.

AMES, K. R.
INTEGRATED VERIFICATION AND TESTING SYSTEM (IVTS) FOR HAL/S PROGRAMS
DOC. DATE: JULY 1983
AUTHORS: SENN, E. H. AMES, K. R. SMITH, K. A.

ANDERSON, G. A.
COMPUTER INTERCONNECTION STRUCTURES: TAXONOMY, CHARACTERISTICS, AND EXAMPLES
DOC. DATE: DECEMBER 1975
AUTHORS: ANDERSON, G. A. JENSEN, E. D.

ANDERSON, T.
FAULT TOLERANCE - PRINCIPLES AND PRACTICE
DOC. DATE: 1981
AUTHORS: ANDERSON, T. LEE, P. A.

(ANDERSON, T. CONTINUED)
PRACTICAL SOFTWARE FAULT TOLERANCE FOR REAL-TIME SYSTEMS. A CLASSIFICATION SYSTEM FOR ERRORS
DOC. DATE: JUNE 1981
AUTHORS: ANDERSON, T. KNIGHT, J.

ARLAT, J.
MODELING OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. DATE: MAY 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.

(ARLAT, J. CONTINUED)
ON THE PERFORMANCE OF SOFTWARE FAULT-TOLERANCE STRATEGIES
DOC. DATE: OCTOBER 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.

(ARLAT, J. CONTINUED)
MODELING AND PERFORMANCE EVALUATION OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. DATE: 1983
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, ALGIRDAS

AVIZIENIS, A.
ON THE IMPLEMENTATION OF N-VERSION PROGRAMMING FOR SOFTWARE FAULT-TOLERANCEDURING PROGRAM EXECUTION
DOC. DATE: NOVEMBER 8, 1977
AUTHORS: AVIZIENIS, A. CHEN, L.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

AVIZIENIS, A. * CONTINUED *

(AVIZIENIS, A. CONTINUED)
N-VERSION PROGRAMMING: A FAULT-TOLERANCE APPROACH TO RELIABILITY OF SOFTWARE OPERATION
DOC. DATE: JUNE 21, 1978
AUTHORS: CHEN, L. AVIZIENIS, A.

(AVIZIENIS, A. CONTINUED)
MODELING OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. DATE: MAY 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.

(AVIZIENIS, A. CONTINUED)
ON THE PERFORMANCE OF SOFTWARE FAULT-TOLERANCE STRATEGIES
DOC. DATE: OCTOBER 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.

(AVIZIENIS, A. CONTINUED)
A UNIFIED RELIABILITY MODEL FOR FAULT-TOLERANT COMPUTERS
DOC. DATE: NOVEMBER 1980
AUTHORS: NG, Y. AVIZIENIS, A.

(AVIZIENIS, A. CONTINUED)
A RELIABILITY AND LIFE-CYCLE EVALUATION TOOL FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: JUNE 1, 1981
AUTHORS: MAKAM, S. AVIZIENIS, A.

(AVIZIENIS, A. CONTINUED)
ARIES 81 - A RELIABILITY AND LIFE CYCLE EVALUATION TOOL FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: JUNE 1982
AUTHORS: MAKAM, S. AVIZIENIS, A.

AVIZIENIS, ALGIRDAS
A SPECIFICATION-ORIENTED MULTI-VERSION SOFTWARE EXPERIMENT
DOC. DATE: 1983
AUTHORS: KELLY, JOHN P. J. AVIZIENIS, ALGIRDAS

(AVIZIENIS, ALGIRDAS CONTINUED)
MODELING AND PERFORMANCE EVALUATION OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. DATE: 1983
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, ALGIRDAS

BACCELLI, F.
ANALYSIS OF M/G/2 - STANDBY REDUNDANT SYSTEM
AUTHORS: BACCELLI, F. TRIVEDI, K.

BAER, JEAN-LOUP
COMPUTER SYSTEMS ARCHITECTURE
DOC. DATE: 1980
AUTHORS: BAER, JEAN-LOUP

BAHRE, REINHARD
THE TRAFFIC FLOW IN A DISTRIBUTED REALTIME COMPUTING SYSTEM (RDC-SYSTEM) WITH A FIBEROPTIC RINGBUS SYSTEM
DOC. DATE: OCTOBER 1981
AUTHORS: HEGER, DIRK BAHRE, REINHARD

BANNISTER, J.
TASK ALLOCATION IN FAULT-TOLERANT DISTRIBUTED SYSTEMS
DOC. DATE: MARCH 1983
AUTHORS: BANNISTER, J. TRIVEDI, K.

BARDIN, B. M.
IMPLEMENTATION OF A REAL-TIME DISTRIBUTED COMPUTER SYSTEM IN ADA
DOC. DATE: 1983
AUTHORS: LANE, D. S. HULING, G. BARDIN, B. M.

BASTANI, F. B.
SOFTWARE RELIABILITY - STATUS AND PERSPECTIVES
DOC. DATE: JULY 1982
AUTHORS: RAMAMOORTHY, C. V. BASTANI, F. B.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

BAVUSO, S. J.
TRENDS IN RELIABILITY MODELING TECHNOLOGY FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: APRIL 1979
AUTHORS: BAVUSO, S. J.

(BAVUSO, S. J. CONTINUED)
METHODOLOGY FOR MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: MCGOUGH, J. G. SWERN, F. BAVUSO, S. J.

(BAVUSO, S. J. CONTINUED)
ADVANCED RELIABILITY MODELING OF FAULT-TOLERANT COMPUTER-BASED SYSTEMS
DOC. DATE: MAY 1982
AUTHORS: BAVUSO, S. J.

BEAUDRY, M.
PERFORMANCE-RELATED RELIABILITY MEASURES FOR COMPUTING
DOC. DATE: JUNE 1978
AUTHORS: BEAUDRY, M.

BEIZER, B.
MICRO-ANALYSIS OF COMPUTER SYSTEM PERFORMANCE
DOC. DATE: 1978
AUTHORS: BEIZER, B.

BERNSTEIN, P. A.
CONCURRENCY CONTROL IN DISTRIBUTED DATABASE SYSTEMS
DOC. DATE: JUNE 1981
AUTHORS: BERNSTEIN, P. A. GOODMAN, N.

BINNS, S. E.
FURTHER DEVELOPMENTS ON THE CAMBRIDGE RING NETWORK AT THE UNIVERSITY OF KENT
DOC. DATE: 1982
AUTHORS: BINNS, S. E. DALLAS, I. N. SPRATT, E. B.

BOOCH, G.
DESCRIBING SOFTWARE DESIGN IN ADA
DOC. DATE: SEPTEMBER 1981
AUTHORS: BOOCH, G.

BOUTE, R. T.
FOUNDATIONS OF THE NEXT-GENERATION MICROPROCESSORS
DOC. DATE: 1981
AUTHORS: BOUTE, R. T.

BOWN, R. L.
A PROPOSAL TO APPLY ADA AND OBJECT ARCHITECTURE TO SATISFY THE DATA REQUIREMENTS OF THE NASA SPACE STATION
DOC. DATE: 1983
AUTHORS: BOWN, R. L. MCKAY, C. W.

(BOWN, R. L. CONTINUED)
ADA AND THE INTEL IAP-432 SUPPORT OF NETWORKS
DOC. DATE: OCTOBER 1983
AUTHORS: MCKAY, C. W. BOWN, R. L. HART, D. A.

BRAMMER, K.
FUNCTIONAL VERSUS COMMUNICATION STRUCTURES IN MODERN AVIONIC SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: BRAMMER, K. WEIMANN, A.

BRANSTAD, M.
VALIDATION, VERIFICATION, AND TESTING OF COMPUTER SOFTWARE
DOC. DATE: JUNE 2, 1982
AUTHORS: ADRION, W. BRANSTAD, M. CHERNIAVSKY, J.

BRIDGEMAN, M.
AUTOMATED RELIABILITY AND FAILURE EFFECTS METHODS FOR DIGITAL FLIGHT CONTROL AND AVIONICS SYSTEMS - VOLUME 1: EVALUATION
DOC. DATE: MARCH 1981
AUTHORS: NESS, W. MCCRARY. W. BRIDGEMAN, M.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

BRIDGEMAN, M. * CONTINUED *

(BRIDGEMAN, M. CONTINUED)
COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS
DOC. DATE: APRIL 1981
AUTHORS: HITT, E. F. BRIDGEMAN, M. ROBINSON, A.

BROCK, LARRY
PROPOSED REQUIREMENTS METHODOLOGY FOR INTEGRATED AVIONICS SYSTEMS
DOC. DATE: JULY 28, 1983
AUTHORS: BROCK, LARRY DEWOLF, J. BARTON

BROOKS, F.P.
THE MYTHICAL MAN-MONTH
DOC. DATE: JULY 1, 1978
AUTHORS: BROOKS, F.P.

BROWNBRIDGE, D.
THE NEWCASTLE CONNECTION OR UNIXES OF THE WORLD UNITE!
DOC. DATE: DECEMBER 1982
AUTHORS: BROWNBRIDGE, D. MARSHALL, L. RANDELL, B.

BRYANT, L.
CARE III FINAL REPORT PHASE I VOLUME I
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.

(BRYANT, L. CONTINUED)
CARE III FINAL REPORT PHASE I VOLUME II
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.

(BRYANT, L. CONTINUED)
CARE III PHASE II REPORT USER'S MANUAL
DOC. DATE: SEPTEMBER 1982
AUTHORS: BRYANT, L. STIFFLER, J.

(BRYANT, L. CONTINUED)
CARE III PHASE II REPORT - MATHEMATICAL DESCRIPTION
DOC. DATE: NOVEMBER 1982
AUTHORS: STIFFLER, J. BRYANT, L.

(BRYANT, L. CONTINUED)
CARE III PHASE III REPORT - TEST AND EVALUATION
DOC. DATE: NOVEMBER 1982
AUTHORS: STIFFLER, J. NEUMANN, J. BRYANT, L.

BUCHAN, P. L.
DARTS: A TOOL FOR SPECIFICATION AND SIMULATION OF REAL-TIME SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: FURTEK, F. C. DEWOLF, J. BARTON BUCHAN, P. L.

BUCKERMAN, S.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IV-ARCHITECTURE RESEARCH FACILITY, ISP DESCRIPTION, SIMULATIONS, DATA COLLECTIONS
DOC. DATE: SEPTEMBER 1977
AUTHORS: GORDON, R. HOWBRIGG, R. BUCKERMAN, S.

BUDDE, D. ET AL
TWO CHIPS ENDOW 32-BIT PROCESSOR WITH FAULT-TOLERANT ARCHITECTURE
DOC. DATE: APRIL 7, 1983
AUTHORS: PETERSON, C. DUZETTE, C. BUDDE, D. ET AL

BUDDE, D. L.
THE EXECUTION UNIT FOR THE VLSI 432 GENERAL DATA PROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: BUDDE, D. L. COLLEY, S. R. DOMENIK, S. L. ET AL

BUHR, R. J. A.
OBJECT-ORIENTED STRUCTURED DESIGN OF LAYERED PROTOCOL SYSTEMS
DOC. DATE: OCTOBER 1982
AUTHORS: BUHR, R. J. A. MICHELL, S.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

BULMAN, DAVID M.
IS ADA THE ANSWER?
AUTHORS: BULMAN, DAVID M.

BURATTI, D.L.
SNEAK SOFTWARE ANALYSIS
DOC. DATE: 1983
AUTHORS: BURATTI, D.L. PINKSTON, W.E. SIMKINS, R.D.

BURR, W.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL III - EVALUATION OF COMPUTER ARCHITECTURE VIA TEST PROGRAMS
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. SHAMAN, P.

(BURR, W. CONTINUED)
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL 1 - INTRODUCTION
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. COLEMAN, A. SMITH, R.

(BURR, W. CONTINUED)
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME II - SELECTION OF CANDIDATE ARCHITECTURES AND INITIAL SCREENING
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. STONE, H.

BUX, W.
A LOCAL-AREA COMMUNICATION NETWORK BASED ON A RELIABLE TOKEN-RING SYSTEM
DOC. DATE: 1982
AUTHORS: BUX, W. CLOSS, F. JANSON, P. A., ET AL

BUZEN, J.
BEST/1 - DESIGN OF A TOOL FOR COMPUTER SYSTEM CAPACITY PLANNING
DOC. DATE: 1979
AUTHORS: BUZEN, J.

CAINE, S. H.
PDL - A TOOL FOR SOFTWARE DESIGN
DOC. DATE: 1975
AUTHORS: CAINE, S. H. GORDON, E. K.

CARNEY, P.
SELECTING ON-BOARD SATELLITE COMPUTER SYSTEMS
DOC. DATE: APRIL 1983
AUTHORS: CARNEY, P.

CASTILLO, X.
A PERFORMANCE - RELIABILITY MODEL FOR COMPUTING SYSTEMS
DOC. DATE: OCTOBER 1980
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.

(CASTILLO, X. CONTINUED)
WORKLOAD, PERFORMANCE AND RELIABILITY OF DIGITAL COMPUTING SYSTEMS
DOC. DATE: JUNE 1981
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.

(CASTILLO, X. CONTINUED)
A WORKLOAD DEPENDENT SOFTWARE RELIABILITY PREDICTION MODEL
DOC. DATE: JUNE 1982
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.

CHAN, W. S.
SPACE PROCESSOR TECHNOLOGY TRADEOFFS
DOC. DATE: 1982
AUTHORS: CRANE, W. W. CHAN, W. S. HUANG, C. C.

CHEN, L.
ON THE IMPLEMENTATION OF N-VERSION PROGRAMMING FOR SOFTWARE FAULT-TOLERANCEDURING PROGRAM EXECUTION
DOC. DATE: NOVEMBER 8, 1977
AUTHORS: AVIZIENIS, A. CHEN, L.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

CHEN, L. * CONTINUED *

(CHEN, L. CONTINUED)
N-VERSION PROGRAMMING: A FAULT-TOLERANCE APPROACH TO RELIABILITY OF SOFTWARE OPERATION
DOC. DATE: JUNE 21, 1978
AUTHORS: CHEN, L. AVIZIENIS, A.

CHERNIAVSKY, J.
VALIDATION, VERIFICATION, AND TESTING OF COMPUTER SOFTWARE
DOC. DATE: JUNE 2, 1982
AUTHORS: ADRION, W. BRANSTAD, M. CHERNIAVSKY, J.

CHEVERS, E. S.
PROGRAM MANAGEMENT OF HIGH TECHNOLOGY SOFTWARE AND HARDWARE FOR THE PROPOSED SPACE STATION
DOC. DATE: OCTOBER 1983
AUTHORS: CHEVERS, E. S.

CLARY, J.
A VALIDATION PROTOTYPE OF SYSTEM RELIABILITY IN LIFE-CRITICAL APPLICATIONS
DOC. DATE: 1980
AUTHORS: TRIVEDI, K. GAULT, J. CLARY, J.

(CLARY, J. CONTINUED)
VALIDATION METHODS RESEARCH FOR FAULT-TOLERANT AVIONICS AND CONTROL SYSTEM SUB-WORKING GROUP MEETING - CARE III PEER REVIEW
DOC. DATE: SEPTEMBER 1980
AUTHORS: TRIVEDI, K. CLARY, J.

CLEARWATERS, A.
COMPUTER FAMILY ARCH SEL COMM-FINAL RPT-V5-PROC FOR AND RESULTS OF THEEVAL UATION OF SOFTWARE BASES OF THE CANDIDATE ARCH FOR THE MIL COMPUTER FAMILY
DOC. DATE: SEPTEMBER 1977
AUTHORS: LIEBLEIN, E. WAGNER, J. CLEARWATERS, A.

(CLEARWATERS, A. CONTINUED)
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME VIII - CFA FINAL SELECTION
DOC. DATE: SEPTEMBER 1977
AUTHORS: CLEARWATERS, A. GORDON, R. SIEWIOREK, DANIEL P.

CLOSS, F.
A LOCAL-AREA COMMUNICATION NETWORK BASED ON A RELIABLE TOKEN-RING SYSTEM
DOC. DATE: 1982
AUTHORS: BUX, W. CLOSS, F. JANSON, P. A., ET AL

COLEMAN, A.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT - VOL VI - LIFE CYCLE COST MODELS
DOC. DATE: SEPTEMBER 1977
AUTHORS: COLEMAN, A. CORNYN, J. SVIRSKY, W.

(COLEMAN, A. CONTINUED)
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL 1 - INTRODUCTION
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. COLEMAN, A. SMITH, R.

COLEMAN, D. M.
A SYSTEMATIC APPROACH TO THE DESIGN OF FAULT TOLERANT DISTRIBUTED COMPUTER SYSTEMS
DOC. DATE: 1983
AUTHORS: COLEMAN, D. M. JORDAN, B. W., JR. SWAMY, S.

COLLEY, S. R.
THE EXECUTION UNIT FOR THE VLSI 432 GENERAL DATA PROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: BUDDE, D. L. COLLEY, S. R. DOMENIK, S. L. ET AL

COLLINSON, R. P. A.
THE CAMBRIDGE RING AND UNIX
DOC. DATE: 1982
AUTHORS: COLLINSON, R. P. A.

CONRAD, T.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IX - A CONSIDERATION OF ISSUES IN THE SELECTION OF A COMPUTER FAMILY ARCHITECTURE
DOC. DATE: SEPTEMBER 1977
AUTHORS: CONRAD, T. ESTELL, R. HAYNES, L.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

CONSTANTINE, L. L.
STRUCTURED DESIGN
DOC. DATE: MAY 1974
AUTHORS: STEVENS, W. P. MEYERS, G. J. CONSTANTINE, L. L.

CORNYN, J.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT - VOL VI - LIFE CYCLE COST MODELS
DOC. DATE: SEPTEMBER 1977
AUTHORS: COLEMAN, A. CORNYN, J. SVIRSKY, W.

COSTES, A.
RELIABILITY AND AVAILABILITY MODELS FOR MAINTAINED SYSTEMS FEATURING HARDWARE FAILURES AND DESIGN FAULTS
DOC. DATE: JUNE 1978
AUTHORS: COSTES, A. LANDRAULT, C. LAPRIE, J.

(COSTES, A. CONTINUED)
SURF - A PROGRAM FOR DEPENDABILITY EVALUATION OF COMPLEX FAULT-TOLERANT COMPUTING SYSTEMS
DOC. DATE: JUNE 1981
AUTHORS: COSTES, A. DOUCET, J. LANDRAULT, C. ET AL

CRANE, W. W.
SPACE PROCESSOR TECHNOLOGY TRADEOFFS
DOC. DATE: 1982
AUTHORS: CRANE, W. W. CHAN, W. S. HUANG, C. C.

CRISTIAN, F.
EXCEPTION HANDLING AND SOFTWARE-FAULT TOLERANCE
DOC. DATE: OCTOBER 1, 1980
AUTHORS: CRISTIAN, F.

CUSTEAU, G.
A SURVEY OF METHODS FOR IMPROVING COMPUTER NETWORK RELIABILITY AND AVAILABILITY
DOC. DATE: NOVEMBER 1977
AUTHORS: MORGAN, D. TAYLOR, P. CUSTEAU, G.

DACOSTA, ROBERT
BABELISM, BABBAGE'S BOOSTER, AND BERNOULLI'S NUMBERS
AUTHORS: DACOSTA, ROBERT

DALLAS, I. N.
FURTHER DEVELOPMENTS ON THE CAMBRIDGE RING NETWORK AT THE UNIVERSITY OF KENT
DOC. DATE: 1982
AUTHORS: BINNS, S. E. DALLAS, I. N. SPRATT, E. B.

DALPEZZO, R. G.
DISTRIBUTED SYSTEM DESIGN DISCIPLINE: AN APPLICATION OF TELL
DOC. DATE: 1981
AUTHORS: RHYNE, J. R. DALPEZZO, R. G.

DALY, K. C.
RELIABILITY AND ACCURACY PREDICTION FOR A REDUNDANT STRAPDOWN NAVIGATOR
DOC. DATE: SEPTEMBER 1981
AUTHORS: HARRISON, J. V. A. DALY, K. C. GAI, E. G.

(DALY, K. C. CONTINUED)
THE IUS PAPERS
DOC. DATE: OCTOBER 1981
AUTHORS: DALY, K. C. HARRISON, J. V. A. GAI, E. G.

DALY, KEVIN C.
INDEPENDENT VERIFICATION AND VALIDATION PROGRAM DEVELOPMENT
DOC. DATE: FEBRUARY 15, 1980
AUTHORS: DALY, KEVIN C.

(DALY, KEVIN C. CONTINUED)
DMSP FLIGHT LOAD PACKAGE CERTIFICATION FOR FLIGHT 6
DOC. DATE: JUNE 30, 1982
AUTHORS: DALY, KEVIN C.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

DAMSKER, D.
TOTALLY DISTRIBUTED, REDUNDANTLY STRUCTURED HARDWARE AND SOFTWARE LOCAL COMPUTER CONTROL NETWORK
DOC. DATE: 1982
AUTHORS: DAMSKER, D.

DAVIS
HMOS PROCESSING IMPROVES RADIATION RESISTANCE OF VLSI MEMORIES AND MICROPROCESSORS
DOC. DATE: NOVEMBER 30, 1982
AUTHORS: DAVIS WOODS ET AL

DELISLE, N. M.
TOOLS FOR SUPPORTING STRUCTURED ANALYSIS
DOC. DATE: 1982
AUTHORS: DELISLE, N. M. MENICOSY, D. E. KERTH, N. L.

DEVERMEYER, B.
A NEW APPROACH FOR NETWORK RELIABILITY ANALYSIS
DOC. DATE: OCTOBER 1982
AUTHORS: DEVERMEYER, B.

DEWOLF, J. BARTON
DARTS: A TOOL FOR SPECIFICATION AND SIMULATION OF REAL-TIME SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: FURTEK, F. C. DEWOLF, J. BARTON BUCHAN, P. L.

(DEWOLF, J. BARTON CONTINUED)
RELIABLE SOFTWARE FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: APRIL 9, 1982
AUTHORS: DEWOLF, J. BARTON FURTEK, F. C. GAI, E. G.

(DEWOLF, J. BARTON CONTINUED)
PROPOSED REQUIREMENTS METHODOLOGY FOR INTEGRATED AVIONICS SYSTEMS
DOC. DATE: JULY 28, 1983
AUTHORS: BROCK, LARRY DEWOLF, J. BARTON

DINIAK, W.
VERIFICATION AND VALIDATION TECHNIQUES APPLIED TO THE RECONFIGURABLE BLOCK-5D SATELLITE SOFTWARE
AUTHORS: DINIAK, W.

DOMENIK, S. L. ET AL
THE EXECUTION UNIT FOR THE VLSI 432 GENERAL DATA PROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: BUDDE, D. L. COLLEY, S. R. DOMENIK, S. L. ET AL

DOUCET, J.
SURF - A PROGRAM FOR DEPENDABILITY EVALUATION OF COMPLEX FAULT-TOLERANT COMPUTING SYSTEMS
DOC. DATE: JUNE 1981
AUTHORS: COSTES, A. DOUCET, J. LANDRAULT, C. ET AL

DOWNES, V.A.
PROGRAMMING EMBEDDED SYSTEMS WITH ADA
DOC. DATE: 1982
AUTHORS: DOWNES, V.A. GOLDSACK, S.J.

DUGAN, J.
DESIGN OF THE HYBRID AUTOMATED RELIABILITY PREDICTOR
AUTHORS: GEIST, R. TRIVEDI, K. DUGAN, J.

DUZETTE, C.
TWO CHIPS ENDOW 32-BIT PROCESSOR WITH FAULT-TOLERANT ARCHITECTURE
DOC. DATE: APRIL 7, 1983
AUTHORS: PETERSON, C. DUZETTE, C. BUDDE, D. ET AL

EHRENFRIED, LT. DANIEL
MIL-STD-1815A (ADA) AND MIL-STD-1750A: PROBLEMS AND SOLUTIONS
DOC. DATE: AUGUST 1983
AUTHORS: EHRENFRIED, LT. DANIEL

ENSLOW, PHILIP H., JR.
MULTIPROCESSORS AND PARALLEL PROCESSING
DOC. DATE: 1974
AUTHORS: ENSLOW, PHILIP H., JR.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

ENSLOW, PHILIP H., JR. * CONTINUED *

(ENSLOW, PHILIP H., JR. CONTINUED)
WHAT IS A "DISTRIBUTED" DATA PROCESSING SYSTEM?
DOC. DATE: JANUARY 1978
AUTHORS: ENSLOW, PHILIP H., JR.

(ENSLOW, PHILIP H., JR. CONTINUED)
DISTRIBUTED AND DECENTRALIZED CONTROL IN FULLY DISTRIBUTED PROCESSING SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: ENSLOW, PHILIP H., JR.

(ENSLOW, PHILIP H., JR. CONTINUED)
DISTRIBUTED DATA PROCESSING --- WHAT IS IT?
DOC. DATE: OCTOBER 1981
AUTHORS: ENSLOW, PHILIP H., JR.

ESTELL, R.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IX - A CONSIDERATION OF ISSUES IN THE SELECTION OF A COMPUTER FAMILY ARCHITECTURE
DOC. DATE: SEPTEMBER 1977
AUTHORS: CONRAD, T. ESTELL, R. HAYNES, L.

ESTRIN, G.
A METHODOLOGY FOR DESIGN OF DIGITAL SYSTEMS - SUPPORTED BY SARA AT THE AGE OF ONE
DOC. DATE: 1978
AUTHORS: ESTRIN, G.

EVANS, A. JR.
COMPARISON OF PROGRAMMING LANGUAGES: ADA, PRAXIS, PASCAL, C
DOC. DATE: 1981
AUTHORS: EVANS, A. JR.

FARBER, R. M.
32-BIT 'MEGAMICRO' EXPLOITS HARDWARE VIRTUAL MEMORY AND 'RAM DISK'
DOC. DATE: OCTOBER 1983
AUTHORS: METCALF, S. M. FARBER, R. M.

FATHI, E.
MULTIPLE MICROPROCESSOR SYSTEMS: WHAT, WHY, AND WHEN
DOC. DATE: MARCH 1983
AUTHORS: FATHI, E. KRIEGER, M.

FELLEMAN, P. G.
AIPS SYSTEM REQUIREMENTS
DOC. DATE: AUGUST 30, 1983
AUTHORS: FELLEMAN, P. G.

FERRARI, D.
WORKLOAD CHARACTERIZATION AND SELECTION IN COMPUTER PERFORMANCE MEASUREMENT
DOC. DATE: AUGUST 7, 1972
AUTHORS: FERRARI, D.

(FERRARI, D. CONTINUED)
COMPUTER SYSTEM PERFORMANCE ANALYSIS
DOC. DATE: 1978
AUTHORS: FERRARI, D.

FEUER, ALAN R.
A COMPARISON OF THE PROGRAMMING LANGUAGES C AND PASCAL
DOC. DATE: MARCH 1982
AUTHORS: FEUER, ALAN R. GEHANI, NARAIN H.

FRIEBELE, E.
RADIATION DAMAGE IN SINGLE-MODE OPTICAL FIBER WAVEGUIDES
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.

(FRIEBELE, E. CONTINUED)
RADIATION DAMAGE OF OPTICAL FIBER WAVEGUIDES AT LONG WAVELENGTHS
DOC. DATE: FEBRUARY 1982
AUTHORS: FRIEBELE, E. GINGERICH, M. LONG, K.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

FRIEBELE, E. * CONTINUED *

(FRIEBELE, E. CONTINUED)
RECOVERY OF RADIATION DAMAGE IN SINGLE-MODE AND MULTIMODE OPTICAL FIBER WAVEGUIDES
DOC. DATE: APRIL 30, 1982
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.

FULLER, S.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL III - EVALUATION OF COMPUTER ARCHITECTURE VIA TEST PROGRAMS
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. SHAMAN, P.

(FULLER, S. CONTINUED)
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME II - SELECTION OF CANDIDATE ARCHITECTURES AND INITIAL SCREENING
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. STONE, H.

FUNK, D.W.
DO-178: ITS HISTORY AND PURPOSE
DOC. DATE: 1983
AUTHORS: FUNK, D.W.

FURCHTGOTT, D.
PERFORMABILITY EVALUATION OF THE SIFT COMPUTER
DOC. DATE: JUNE 1980
AUTHORS: MEYER, J. F. FURCHTGOTT, D. WU, L.

(FURCHTGOTT, D. CONTINUED)
A BIBLIOGRAPHY ON FORMAL METHODS FOR SYSTEM SPECIFICATION, DESIGN AND VALIDATION
DOC. DATE: JAN 1982
AUTHORS: MEYER, J. F. FURCHTGOTT, D. MOVAGHAR, A.

(FURCHTGOTT, D. CONTINUED)
PERFORMABILITY EVALUATION OF COMPUTING SYSTEMS USING REWARD MODELS
DOC. DATE: AUGUST 1983
AUTHORS: FURCHTGOTT, D. MEYER, J. F.

FURTEK, F. C.
DARTS: A TOOL FOR SPECIFICATION AND SIMULATION OF REAL-TIME SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: FURTEK, F. C. DEWOLF, J. BARTON BUCHAN, P. L.

(FURTEK, F. C. CONTINUED)
RELIABLE SOFTWARE FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: APRIL 9, 1982
AUTHORS: DEWOLF, J. BARTON FURTEK, F. C. GAI, E. G.

GAI, E. G.
RELIABILITY AND ACCURACY PREDICTION FOR A REDUNDANT STRAPDOWN NAVIGATOR
DOC. DATE: SEPTEMBER 1981
AUTHORS: HARRISON, J. V. A. DALY, K. C. GAI, E. G.

(GAI, E. G. CONTINUED)
THE IUS PAPERS
DOC. DATE: OCTOBER 1981
AUTHORS: DALY, K. C. HARRISON, J. V. A. GAI, E. G.

(GAI, E. G. CONTINUED)
RELIABLE SOFTWARE FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: APRIL 9, 1982
AUTHORS: DEWOLF, J. BARTON FURTEK, F. C. GAI, E. G.

(GAI, E. G. CONTINUED)
MEASURES OF MERIT FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: 1983
AUTHORS: GAI, E. G. ADAMS, M.

GANESH, S. L.
ISSUES IN SOFTWARE RELIABILITY
DOC. DATE: JULY 1981
AUTHORS: RAMAMOORTHY, C. V. GANESH, S. L.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

GAULT, J.
A VALIDATION PROTOTYPE OF SYSTEM RELIABILITY IN LIFE-CRITICAL APPLICATIONS
DOC. DATE: 1980
AUTHORS: TRIVEDI, K. GAULT, J. CLARY, J.

GAY, F.
PERFORMANCE EVALUATION OF GRACEFULLY DEGRADING SYSTEMS
DOC. DATE: JUNE 1979
AUTHORS: GAY, F. KELETSON, M.

GEHANI, NARAIN
ADA -- AN ADVANCED INTRODUCTION
DOC. DATE: 1983
AUTHORS: GEHANI, NARAIN

GEHANI, NARAIN H.
A COMPARISON OF THE PROGRAMMING LANGUAGES C AND PASCAL
DOC. DATE: MARCH 1982
AUTHORS: FEUER, ALAN R. GEHANI, NARAIN H.

GEIST, R.
DECOMPOSITION IN RELIABILITY ANALYSIS OF FAULT-TOLERANT SYSTEMS
AUTHORS: TRIVEDI, K. GEIST, R.

(GEIST, R. CONTINUED)
DESIGN OF THE HYBRID AUTOMATED RELIABILITY PREDICTOR
AUTHORS: GEIST, R. TRIVEDI, K. DUGAN, J.

(GEIST, R. CONTINUED)
THE INTEGRATION OF USER PERCEPTION IN THE HETEROGENEOUS M/M/2 QUEUE
AUTHORS: GEIST, R. TRIVEDI, K.

(GEIST, R. CONTINUED)
OPTIMAL DESIGN OF MULTILEVEL STORAGE HIERARCHIES
DOC. DATE: MARCH 1982
AUTHORS: GEIST, R. TRIVEDI, K.

(GEIST, R. CONTINUED)
ULTRA-HIGH RELIABILITY PREDICTION FOR FAULT-TOLERANT COMPUTER SYSTEMS
DOC. DATE: FEBRUARY 1983
AUTHORS: GEIST, R. TRIVEDI, K.

GHEZZI, CARLO
PROGRAMMING LANGUAGE CONCEPTS
DOC. DATE: 1982
AUTHORS: GHEZZI, CARLO JAZAYERI, MEHDI

GILBERT, LUCY
SUITABILITY OF ADA TASKS FOR DISTRIBUTED COMPUTING
DOC. DATE: SEPTEMBER 1, 1983
AUTHORS: GILBERT, LUCY

GINGERICH, M.
RADIATION DAMAGE IN SINGLE-MODE OPTICAL FIBER WAVEGUIDES
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.

(GINGERICH, M. CONTINUED)
RADIATION DAMAGE OF OPTICAL FIBER WAVEGUIDES AT LONG WAVELENGTHS
DOC. DATE: FEBRUARY 1982
AUTHORS: FRIEBELE, E. GINGERICH, M. LONG, K.

(GINGERICH, M. CONTINUED)
RECOVERY OF RADIATION DAMAGE IN SINGLE-MODE AND MULTIMODE OPTICAL FIBER WAVEGUIDES
DOC. DATE: APRIL 30, 1982
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.

GMEINER, L.
SOFTWARE DIVERSITY IN REACTOR PROTECTION SYSTEMS: AN EXPERIMENT
DOC. DATE: 1979
AUTHORS: GMEINER, L. VOGES, U.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

GOLDSACK, S.J.
PROGRAMMING EMBEDDED SYSTEMS WITH ADA
DOC. DATE: 1982
AUTHORS: DOWNES, V.A. GOLDSACK, S.J.

GOODMAN, N.
CONCURRENCY CONTROL IN DISTRIBUTED DATABASE SYSTEMS
DOC. DATE: JUNE 1981
AUTHORS: BERNSTEIN, P. A. GOODMAN, N.

GORDON, E. K.
PDL - A TOOL FOR SOFTWARE DESIGN
DOC. DATE: 1975
AUTHORS: CAINE, S. H. GORDON, E. K.

GORDON, R.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME VIII - CFA FINAL SELECTION
DOC. DATE: SEPTEMBER 1977
AUTHORS: CLEARWATERS, A. GORDON, R. SIEWIOREK, DANIEL P.

(GORDON, R. CONTINUED)
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IV-ARCHITECTURE RESEARCH FACILITY, ISP DESCRIPTION, SIMULATIONS, DATA COLLECTIONS
DOC. DATE: SEPTEMBER 1977
AUTHORS: GORDON, R. HOWBRIGG, R. BUCKERMAN, S.

GRIMSDALE, R. L.
SOME EXPERIENCES OF IMPLEMENTING THE ADA CONCURRENCY FACILITIES ON A DISTRIBUTED MULTIPROCESSOR COMPUTER SYSTEM
DOC. DATE: OCTOBER 1982
AUTHORS: SHOJA, G. C. HALSALL, F. GRIMSDALE, R. L.

GRNAROV, A.
MODELING OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. DATE: MAY 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.

(GRNAROV, A. CONTINUED)
ON THE PERFORMANCE OF SOFTWARE FAULT-TOLERANCE STRATEGIES
DOC. DATE: OCTOBER 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.

(GRNAROV, A. CONTINUED)
MODELING AND PERFORMANCE EVALUATION OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. DATE: 1983
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, ALGIRDAS

GUCCIONE, L.
CARE III FINAL REPORT PHASE I VOLUME I
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.

(GUCCIONE, L. CONTINUED)
CARE III FINAL REPORT PHASE I VOLUME II
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.

GUILLEMONT, M.
THE CHORUS DISTRIBUTED OPERATING SYSTEM: DESIGN AND IMPLEMENTATION
DOC. DATE: 1982
AUTHORS: GUILLEMONT, M.

HAGSTROM, J.
A NEW ALGORITHM FOR THE RELIABILITY ANALYSIS OF MULTI-TERMINAL NETWORKS
DOC. DATE: OCTOBER 1981
AUTHORS: SATYANAVAYANA, A. HAGSTROM, J.

HALSALL, F.
SOME EXPERIENCES OF IMPLEMENTING THE ADA CONCURRENCY FACILITIES ON A DISTRIBUTED MULTIPROCESSOR COMPUTER SYSTEM
DOC. DATE: OCTOBER 1982
AUTHORS: SHOJA, G. C. HALSALL, F. GRIMSDALE, R. L.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

HARRISON, J. V. A.
RELIABILITY AND ACCURACY PREDICTION FOR A REDUNDANT STRAPDOWN NAVIGATOR
DOC. DATE: SEPTEMBER 1981
AUTHORS: HARRISON, J. V. A. DALY, K. C. GAI, E. G.

(HARRISON, J. V. A. CONTINUED)
THE IUS PAPERS
DOC. DATE: OCTOBER 1981
AUTHORS: DALY, K. C. HARRISON, J. V. A. GAI, E. G.

HART, D. A.
ADA AND THE INTEL IAP-432 SUPPORT OF NETWORKS
DOC. DATE: OCTOBER 1983
AUTHORS: MCKAY, C. W. BOWN, R. L. HART, D. A.

(HART, D. A. CONTINUED)
THE EFFECTS OF ADA AND THE INTEL IAPX432 UPON THE MANAGEMENT OF COMPUTER RESOURCES
DOC. DATE: OCTOBER 1983
AUTHORS: HART, D. A.

HART, L.
GENERALIZED COMPUTER SYSTEMS SIMULATOR II (GCSS II) USER'S GUIDE
AUTHORS: HART, L.

HATAYAMA, K.
PERFORMANCE EVALUATION OF A FAULT-TOLERANT COMPUTING SYSTEM
DOC. DATE: JUNE 1979
AUTHORS: MINE, H. HATAYAMA, K.

HAYNES, L.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IX - A CONSIDERATION OF ISSUES IN THE SELECTION OF A COMPUTER FAMILY ARCHITECTURE
DOC. DATE: SEPTEMBER 1977
AUTHORS: CONRAD, T. ESTELL, R. HAYNES, L.

HECHT, H.
FAULT-TOLERANT COMPUTING
DOC. DATE: JULY 1, 1979
AUTHORS: HECHT, H.

(HECHT, H. CONTINUED)
USE OF FAULT TREES FOR THE DESIGN OF RECOVERY BLOCKS
DOC. DATE: JUNE 22, 1982
AUTHORS: HECHT, H. HECHT, M.

(HECHT, H. CONTINUED)
CURRENT ISSUES IN FAULT-TOLERANT COMPUTING
DOC. DATE: OCTOBER 1, 1983
AUTHORS: HECHT, H.

HECHT, HERBERT
FAULT-TOLERANT SOFTWARE
DOC. DATE: AUGUST 1979
AUTHORS: HECHT, HERBERT

HECHT, M.
USE OF FAULT TREES FOR THE DESIGN OF RECOVERY BLOCKS
DOC. DATE: JUNE 22, 1982
AUTHORS: HECHT, H. HECHT, M.

HEGER, DIRK
THE TRAFFIC FLOW IN A DISTRIBUTED REALTIME COMPUTING SYSTEM (RDC-SYSTEM) WITH A FIBEROPTIC RINGBUS SYSTEM
DOC. DATE: OCTOBER 1981
AUTHORS: HEGER, DIRK BAHRE, REINHARD

HEIDELBERGER, P.
ANALYTIC QUEUEING THEORY FOR PROGRAMS WITH INTERNAL CONCURRENCY
DOC. DATE: 1983
AUTHORS: HEIDELBERGER, P. TRIVEDI, K.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

HEINSELMAN, R. C.
SYSTEM DESIGN SELECTION FOR DISTRIBUTED DATA SYSTEMS
AUTHORS: HEINSELMAN, R. C.

HENDRICKS, H.
WAVELENGTH DIVISION MULTIPLEXING FOR FUTURE SPACE STATION DATA SYSTEMS
DOC. DATE: AUGUST 25, 1983
AUTHORS: HENDRICKS, H. MURRAY, N.

HENINGER, K. L. ET AL
SPECIFYING SOFTWARE REQUIREMENTS FOR COMPLEX SYSTEMS: NEW TECHNIQUES AND THEIR APPLICATION
DOC. DATE: 1980
AUTHORS: HENINGER, K. L. ET AL

HENRY, G.
COST, PERFORMANCE IMPROVE AS FIBER-OPTICS USE GROWS
DOC. DATE: AUGUST 15, 1983
AUTHORS: HENRY, G.

HENRY, R.
MANAGEMENT GUIDE FOR INDEPENDENT VERIFICATION AND VALIDATION (IV&V)
DOC. DATE: AUGUST 1980
AUTHORS: HENRY, R.

HENRY, S.
SOFTWARE STRUCTURE METRICS BASED ON INFORMATION FLOW
DOC. DATE: SEPTEMBER 1981
AUTHORS: HENRY, S. KAFURA, D.

HERNANDEZ, C.M.
ADVANCED AVIONICS SYSTEM FOR AN AEROSPACE PLANE
DOC. DATE: 1983
AUTHORS: JORGENSEN, D.E. HERNANDEZ, C.M.

HERSHEY, E. A.
PSL/PSA: A COMPUTER AIDED TECHNIQUE FOR STRUCTURED DOCUMENTATION AND ANALYSIS OF INFORMATION PROCESSING
DOC. DATE: JANUARY 1977
AUTHORS: TEICHROEW, D. HERSHEY, E. A.

HINGARH, H.
16-BIT BIPOLAR MICROPROCESSOR MARCHES TO STANDARD INSTRUCTION SET
DOC. DATE: APRIL 7, 1983
AUTHORS: MOR, S. HINGARH, H. VORA, M. ET AL

HITT, E. F.
COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS
DOC. DATE: APRIL 1981
AUTHORS: HITT, E. F. BRIDGEMAN, M. ROBINSON, A.

HOPKINS, ALBERT L.
SURVIVAL AND DISPATCH PROBABILITY MODELS FOR THE FTMP COMPUTER
DOC. DATE: JUNE 1978
AUTHORS: LALA, JAYNARAYAN H. HOPKINS, ALBERT L.

(HOPKINS, ALBERT L. CONTINUED)
FTMP - A HIGHLY RELIABLE FAULT-TOLERANT MULTIPROCESSOR FOR AIRCRAFT
DOC. DATE: OCTOBER 1978
AUTHORS: HOPKINS, ALBERT L. SMITH, T. BASIL, III LALA, JAYNARAYAN H.

HOUGHTON, R. C.
SOFTWARE DEVELOPMENT TOOLS
DOC. DATE: MARCH 1982
AUTHORS: HOUGHTON, R. C.

HOUGHTON, R.C.
A TAXONOMY OF TOOL FEATURES FOR THE ADA PROGRAMMING SUPPORT ENVIRONMENT (APSE)
AUTHORS: HOUGHTON, R.C.

HOWBRIGG, R.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IV-ARCHITECTURE RESEARCH FACILITY, ISP DESCRIPTION, SIMULATIONS, DATA COLLECTIONS
DOC. DATE: SEPTEMBER 1977
AUTHORS: GORDON, R. HOWBRIGG, R. BUCKERMAN, S.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

HOWDEN, W.
FUNCTIONAL PROGRAM TESTING
DOC. DATE: MARCH 1980
AUTHORS: HOWDEN, W.

(HOWDEN, W. CONTINUED)
COMPLETENESS CRITERIA FOR TESTING ELEMENTARY PROGRAM FUNCTIONS
DOC. DATE: SEPTEMBER 1981
AUTHORS: HOWDEN, W.

HUANG, C. C.
SPACE PROCESSOR TECHNOLOGY TRADEOFFS
DOC. DATE: 1982
AUTHORS: CRANE, W. W. CHAN, W. S. HUANG, C. C.

HULING, G.
IMPLEMENTATION OF A REAL-TIME DISTRIBUTED COMPUTER SYSTEM IN ADA
DOC. DATE: 1983
AUTHORS: LANE, D. S. HULING, G. BARDIN, B. M.

INGLE, A.
RELIABILITY MODELS FOR MULTIPROCESSOR SYSTEMS WITH AND WITHOUT PERIODIC MAINTENANCE
DOC. DATE: JUNE 1977
AUTHORS: INGLE, A. SIEWIOREK, DANIEL P.

JACKSON, M. A.
PRINCIPLES OF PROGRAM DESIGN
DOC. DATE: 1975
AUTHORS: JACKSON, M. A.

JANSON, P. A., ET AL
A LOCAL-AREA COMMUNICATION NETWORK BASED ON A RELIABLE TOKEN-RING SYSTEM
DOC. DATE: 1982
AUTHORS: BUX, W. CLOSS, F. JANSON, P. A., ET AL

JAZAYERI, MEHDI
PROGRAMMING LANGUAGE CONCEPTS
DOC. DATE: 1982
AUTHORS: GHEZZI, CARLO JAZAYERI, MEHDI

JENSEN, E. D.
COMPUTER INTERCONNECTION STRUCTURES: TAXONOMY, CHARACTERISTICS, AND EXAMPLES
DOC. DATE: DECEMBER 1975
AUTHORS: ANDERSON, G. A. JENSEN, E. D.

JORDAN, B. W., JR.
A SYSTEMATIC APPROACH TO THE DESIGN OF FAULT TOLERANT DISTRIBUTED COMPUTER SYSTEMS
DOC. DATE: 1983
AUTHORS: COLEMAN, D. M. JORDAN, B. W., JR. SWAMY, S.

JORGENSEN, D.E.
ADVANCED AVIONICS SYSTEM FOR AN AEROSPACE PLANE
DOC. DATE: 1983
AUTHORS: JORGENSEN, D.E. HERNANDEZ, C.M.

KAFURA, D.
SOFTWARE STRUCTURE METRICS BASED ON INFORMATION FLOW
DOC. DATE: SEPTEMBER 1981
AUTHORS: HENRY, S. KAFURA, D.

KEINZLE, M.
A SYSTEMATIC APPROACH TO THE PERFORMANCE MODELING OF COMPUTER SYSTEMS
DOC. DATE: 1979
AUTHORS: KEINZLE, M. SEVEIK, K.

KELETSON, M.
PERFORMANCE EVALUATION OF GRACEFULLY DEGRADING SYSTEMS
DOC. DATE: JUNE 1979
AUTHORS: GAY, F. KELETSON, M.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

KELLY, JOHN P. J.
A SPECIFICATION-ORIENTED MULTI-VERSION SOFTWARE EXPERIMENT
DOC. DATE: 1983
AUTHORS: KELLY, JOHN P. J. AVIZIENIS, ALGIRDAS

KERNIGHAN, BRIAN
THE C PROGRAMMING LANGUAGE
DOC. DATE: 1978
AUTHORS: KERNIGHAN, BRIAN RITCHIE, DENNIS

KERNIGHAN, BRIAN W.
WHY PASCAL IS NOT MY FAVORITE PROGRAMMING LANGUAGE
DOC. DATE: JULY 18, 1981
AUTHORS: KERNIGHAN, BRIAN W.

KERTH, N. L.
TOOLS FOR SUPPORTING STRUCTURED ANALYSIS
DOC. DATE: 1982
AUTHORS: DELISLE, N. M. MENICOSY, D. E. KERTH, N. L.

KEZUNOVIC, M.
MICROPROCESSOR SYSTEMS AND ARCHITECTURES FOR APPLICATIONS TO THE CONTROL AND PROTECTION OF ELECTRIC POWER SYSTEMS
DOC. DATE: 1979
AUTHORS: RUMMER, D. I. KEZUNOVIC, M.

KINICKI, R.
A MODEL FOR COMPUTER CONFIGURATION DESIGN
DOC. DATE: APRIL 1980
AUTHORS: TRIVEDI, K. KINICKI, R.

KNIGHT, J.
PRACTICAL SOFTWARE FAULT TOLERANCE FOR REAL-TIME SYSTEMS. A CLASSIFICATION SYSTEM FOR ERRORS
DOC. DATE: JUNE 1981
AUTHORS: ANDERSON, T. KNIGHT, J.

KOHLER, W. H.
A SURVEY OF TECHNIQUES FOR SYNCHRONIZATION AND RECOVERY IN DECENTRALIZED COMPUTER SYSTEMS
DOC. DATE: JUNE 1, 1981
AUTHORS: KOHLER, W. H.

KREUTZER, W.
COMPUTER SYSTEM MODELING AND SIMULATION
DOC. DATE: NOVEMBER 1979
AUTHORS: KREUTZER, W.

KRIEGER, M.
MULTIPLE MICROPROCESSOR SYSTEMS: WHAT, WHY, AND WHEN
DOC. DATE: MARCH 1983
AUTHORS: FATHI, E. KRIEGER, M.

KRISHNA, C.
A UNIFIED METHOD FOR EVALUATING REAL-TIME COMPUTER CONTROLLERS: A CASE STUDY
DOC. DATE: 1983
AUTHORS: SHINK, G. KRISHNA, C. LEE, Y.

LALA, JAYNARAYAN H.
SURVIVAL AND DISPATCH PROBABILITY MODELS FOR THE FTMP COMPUTER
DOC. DATE: JUNE 1978
AUTHORS: LALA, JAYNARAYAN H. HOPKINS, ALBERT L.

(LALA, JAYNARAYAN H. CONTINUED)
FTMP - A HIGHLY RELIABLE FAULT-TOLERANT MULTIPROCESSOR FOR AIRCRAFT
DOC. DATE: OCTOBER 1978
AUTHORS: HOPKINS, ALBERT L. SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(LALA, JAYNARAYAN H. CONTINUED)
MARK1 - MARKOV MODELING PACKAGE
DOC. DATE: MARCH 1983
AUTHORS: LALA, JAYNARAYAN H.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

LALA, JAYNARAYAN H. * CONTINUED *

(LALA, JAYNARAYAN H. CONTINUED)
DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER, VOLUME I, FTMP PRINCIPLES OF OPERATION
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(LALA, JAYNARAYAN H. CONTINUED)
DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER, VOLUME II, FTMP SOFTWARE
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(LALA, JAYNARAYAN H. CONTINUED)
DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER, VOLUME III, FTMP TEST AND EVALUATION
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(LALA, JAYNARAYAN H. CONTINUED)
FAULT DETECTION, ISOLATION AND RECONFIGURATION IN FTMP: METHODS AND EXPERIMENTAL RESULTS
DOC. DATE: NOVEMBER 2, 1983
AUTHORS: LALA, JAYNARAYAN H.

LAMPORT, L.
TIME, CLOCKS, AND THE ORDERING OF EVENTS IN A DISTRIBUTED SYSTEM
DOC. DATE: JULY 1978
AUTHORS: LAMPORT, L.

LANDRAULT, C.
SURF - A PROGRAM FOR MODELING AND RELIABILITY PREDICTION FOR FAULT-TOLERANTCOMPUTING SYSTEMS
DOC. DATE: 1978
AUTHORS: LANDRAULT, C. LAPRIE, J.

(LANDRAULT, C. CONTINUED)
RELIABILITY AND AVAILABILITY MODELS FOR MAINTAINED SYSTEMS FEATURING HARDWARE FAILURES AND DESIGN FAULTS
DOC. DATE: JUNE 1978
AUTHORS: COSTES, A. LANDRAULT, C. LAPRIE, J.

LANDRAULT, C. ET AL
SURF - A PROGRAM FOR DEPENDABILITY EVALUATION OF COMPLEX FAULT-TOLERANT COMPUTING SYSTEMS
DOC. DATE: JUNE 1981
AUTHORS: COSTES, A. DOUCET, J. LANDRAULT, C. ET AL

LANDROW, H.
RELIABILITY AND AVAILABILITY MODELING OF SYSTEMS FEATURING HARDWARE AND SOFTWARE FAULTS
DOC. DATE: JUNE 1977
AUTHORS: LANDROW, H. LAPIE, J.

LANE, D.
MODELS AND MEASUREMENTS OF PARALLELISM FOR A DISTRIBUTED COMPUTER SYSTEM
AUTHORS: LANE, D.

LANE, D. S.
IMPLEMENTATION OF A REAL-TIME DISTRIBUTED COMPUTER SYSTEM IN ADA
DOC. DATE: 1983
AUTHORS: LANE, D. S. HULING, G. BARDIN, B. M.

LAPIE, J.
RELIABILITY AND AVAILABILITY MODELING OF SYSTEMS FEATURING HARDWARE AND SOFTWARE FAULTS
DOC. DATE: JUNE 1977
AUTHORS: LANDROW, H. LAPIE, J.

LAPRIE, J.
SURF - A PROGRAM FOR MODELING AND RELIABILITY PREDICTION FOR FAULT-TOLERANTCOMPUTING SYSTEMS
DOC. DATE: 1978
AUTHORS: LANDRAULT, C. LAPRIE, J.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

LAPRIE, J. * CONTINUED *

(LAPRIE, J. CONTINUED)
RELIABILITY AND AVAILABILITY MODELS FOR MAINTAINED SYSTEMS FEATURING HARDWARE FAILURES AND DESIGN FAULTS
DOC. DATE: JUNE 1978
AUTHORS: COSTES, A. LANDRAULT, C. LAPRIE, J.

LEACH, DANIEL M.
IMPLEMENTATION LANGUAGES FOR DATA ABSTRACTIONS
DOC. DATE: MARCH 19, 1984
AUTHORS: LEACH, DANIEL M. SATKO, JAMES E.

LEE, P. A.
FAULT TOLERANCE - PRINCIPLES AND PRACTICE
DOC. DATE: 1981
AUTHORS: ANDERSON, T. LEE, P. A.

LEE, Y.
A UNIFIED METHOD FOR EVALUATING REAL-TIME COMPUTER CONTROLLERS: A CASE STUDY
DOC. DATE: 1983
AUTHORS: SHINK, G. KRISHNA, C. LEE, Y.

LEEPER, K.R.
THE MISSING LINK FOR ADVANCED AVIONICS SYSTEMS EXECUTIVES
DOC. DATE: 1983
AUTHORS: LEEPER, K.R.

LIAN, R. C.
DESIGN OF A NETWORK OPERATING SYSTEM FOR THE DISTRIBUTED DOUBLE-LOOP COMPUTER NETWORK (DDLCN)
DOC. DATE: 1982
AUTHORS: LIU, M. T. TSAY, D. P. LIAN, R. C.

LIEBLEIN, E.
COMPUTER FAMILY ARCH SEL COMM-FINAL RPT-V5-PROC FOR AND RESULTS OF THEEVAL UATION OF SOFTWARE BASES OF THE CANDIDATE ARCH FOR THE MIL COMPUTER FAMILY
DOC. DATE: SEPTEMBER 1977
AUTHORS: LIEBLEIN, E. WAGNER, J. CLEARWATERS, A.

LIU, M. T.
DESIGN OF A NETWORK OPERATING SYSTEM FOR THE DISTRIBUTED DOUBLE-LOOP COMPUTER NETWORK (DDLCN)
DOC. DATE: 1982
AUTHORS: LIU, M. T. TSAY, D. P. LIAN, R. C.

LONG, D.
RADIATION HARDNESS OF NEW TECHNOLOGIES - STATE OF THE ART REVIEW
DOC. DATE: NOVEMBER 30, 1982
AUTHORS: LONG. D.

LONG, K.
RADIATION DAMAGE IN SINGLE-MODE OPTICAL FIBER WAVEGUIDES
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.

(LONG, K. CONTINUED)
RADIATION DAMAGE OF OPTICAL FIBER WAVEGUIDES AT LONG WAVELENGTHS
DOC. DATE: FEBRUARY 1982
AUTHORS: FRIEBELE, E. GINGERICH, M. LONG, K.

(LONG, K. CONTINUED)
RECOVERY OF RADIATION DAMAGE IN SINGLE-MODE AND MULTIMODE OPTICAL FIBER WAVEGUIDES
DOC. DATE: APRIL 30, 1982
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.

LUPPOLD, R.
RELIABILITY AND AVAILABILITY MODELS FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: AUGUST 1983
AUTHORS: LUPPOLD, R.

MA, Y. W.
DESIGN AND ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS
DOC. DATE: 1981
AUTHORS: RAMAMOORTHY, C. V. MA, Y. W.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

MAKAM, S.
A RELIABILITY AND LIFE-CYCLE EVALUATION TOOL FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: JUNE 1, 1981
AUTHORS: MAKAM, S. AVIZIENIS, A.

(MAKAM, S. CONTINUED)
ARIES 81 - A RELIABILITY AND LIFE CYCLE EVALUATION TOOL FOR FAULT-TOLERANT SYSTEMS
DOC. DATE: JUNE 1982
AUTHORS: MAKAM, S. AVIZIENIS, A.

MARCHBANKS, M.
THE ANALYSIS OF THE STATISTICAL AND HISTORICAL INFORMATION GATHERED DURING THE DEVELOPMENT OF THE SHUTTLE ORBITER PRIMARY FLIGHT SOFTWARE
DOC. DATE: JUNE 1982
AUTHORS: SIMMONS, D. MARCHBANKS, M. QUICK, M.

MARINOS, P.
A SIMULATOR FOR RELIABILITY PREDICTIONS OF FAULT-TOLERANT SYSTEM ARCHITECTURES
DOC. DATE: SEPTEMBER 1982
AUTHORS: MARINOS, P.

MARSHALL, L.
THE NEWCASTLE CONNECTION OR UNIXES OF THE WORLD UNITE!
DOC. DATE: DECEMBER 1982
AUTHORS: BROWNBRIDGE, D. MARSHALL, L. RANDELL, B.

MARTIN, J. T.
FUNCTIONAL DOCUMENTATION - A PRACTICAL AID TO THE ORDERLY SOLUTION OF THE SYSTEM DESIGN PROBLEM
DOC. DATE: OCTOBER 1981
AUTHORS: MARTIN, J. T.

MATHUR, F.
RELIABILITY MODELING AND ARCHITECTURE OF ULTRA-RELIABLE FAULT-TOLERANT DIGITAL COMPUTERS
DOC. DATE: 1970
AUTHORS: MATHUR, F.

MCCALLA, G. I.
AN APPROACH TO THE ORGANIZATION OF KNOWLEDGE AND ITS USE IN NATURAL LANGUAGE RECALL TASKS
DOC. DATE: 1983
AUTHORS: MCCALLA, G. I.

MCCONNEL, S. R.
THE MEASUREMENT AND ANALYSIS OF TRANSIENT ERRORS IN DIGITAL COMPUTER SYSTEMS
DOC. DATE: JUNE 1979
AUTHORS: MCCONNEL, S. R. SIEWIOREK, DANIEL P.

MCCRARY, W.
AUTOMATED RELIABILITY AND FAILURE EFFECTS METHODS FOR DIGITAL FLIGHT CONTROL AND AVIONICS SYSTEMS - VOLUME 1: EVALUATION
DOC. DATE: MARCH 1981
AUTHORS: NESS, W. MCCRARY, W. BRIDGEMAN, M.

MCCUEN, J. W.
NEXT GENERATION MILITARY AIRCRAFT WILL REQUIRE HIERARCHICAL/MULTILEVEL INFORMATION TRANSFER SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: MCCUEN, J. W.

MCDONALD, W.
A FLEXIBLE DISTRIBUTED TESTBED FOR REAL-TIME APPLICATIONS
DOC. DATE: OCTOBER 1982
AUTHORS: MCDONALD, W. SMITH, R.

MCGOUGH, J. G.
MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: MCGOUGH, J. G. SWERN, F.

(MCGOUGH, J. G. CONTINUED)
METHODOLOGY FOR MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: MCGOUGH, J. G. SWERN, F. BAVUSO, S. J.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

MCGOUGH, J. G. * CONTINUED *

(MCGOUGH, J. G. CONTINUED)
MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR - PART II
DOC. DATE: 1983
AUTHORS: MCGOUGH, J. G. SWERN, F.

MCKAY, C. W.
A PROPOSAL TO APPLY ADA AND OBJECT ARCHITECTURE TO SATISFY THE DATA REQUIREMENTS OF THE NASA SPACE STATION
DOC. DATE: 1983
AUTHORS: BOWN, R. L. MCKAY, C. W.

(MCKAY, C. W. CONTINUED)
SPACE STATION INFORMATION SYSTEMS
DOC. DATE: 1983
AUTHORS: SWINGLE, W. J. MCKAY, C. W.

(MCKAY, C. W. CONTINUED)
ADA AND THE INTEL IAP-432 SUPPORT OF NETWORKS
DOC. DATE: OCTOBER 1983
AUTHORS: MCKAY, C. W. BOWN, R. L. HART, D. A.

MCKELVEY, T. R.
DESIGN OF SOFTWARE FOR DISTRIBUTED/MULTIPROCESSOR SYSTEMS
DOC. DATE: 1982
AUTHORS: MCKELVEY, T. R. AGRAWAL, D. P.

MELLIAR-SMITH, P.
FORMAL SPECIFICATION AND MECHANICAL VERIFICATION OF SIFT: A FAULT-TOLERANT FLIGHT CONTROL SYSTEM
DOC. DATE: JULY 1982
AUTHORS: MELLIAR-SMITH, P. SCHWARTZ, R.

MENICOSY, D. E.
TOOLS FOR SUPPORTING STRUCTURED ANALYSIS
DOC. DATE: 1982
AUTHORS: DELISLE, N. M. MENICOSY, D. E. KERTH, N. L.

METCALF, S. M.
32-BIT 'MEGAMICRO' EXPLOITS HARDWARE VIRTUAL MEMORY AND 'RAM DISK'
DOC. DATE: OCTOBER 1983
AUTHORS: METCALF, S. M. FARBER, R. M.

MEYER, J. F.
PERFORMABILITY EVALUATION OF THE SIFT COMPUTER
DOC. DATE: JUNE 1980
AUTHORS: MEYER, J. F. FURCHTGOTT, D. WU, L.

(MEYER, J. F. CONTINUED)
ON EVALUATING THE PERFORMABILITY OF DEGRADABLE COMPUTING SYSTEMS
DOC. DATE: AUGUST 1980
AUTHORS: MEYER, J. F.

(MEYER, J. F. CONTINUED)
A BIBLIOGRAPHY ON FORMAL METHODS FOR SYSTEM SPECIFICATION, DESIGN AND VALIDATION
DOC. DATE: JAN 1982
AUTHORS: MEYER, J. F. FURCHTGOTT, D. MOVAGHAR, A.

(MEYER, J. F. CONTINUED)
CLOSED FORM SOLUTION OF PERFORMABILITY
DOC. DATE: JULY 1982
AUTHORS: MEYER, J. F.

(MEYER, J. F. CONTINUED)
MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS
DOC. DATE: JULY 1982
AUTHORS: MEYER, J. F.

(MEYER, J. F. CONTINUED)
PERFORMABILITY EVALUATION OF COMPUTING SYSTEMS USING REWARD MODELS
DOC. DATE: AUGUST 1983
AUTHORS: FURCHTGOTT, D. MEYER, J. F.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

MEYER, J. F. * CONTINUED *

(MEYER, J. F. CONTINUED)
PERFORMABILITY MODELING OF DISTRIBUTED REAL-TIME SYSTEMS
DOC. DATE: AUGUST 1983
AUTHORS: MEYER, J. F.

MEYERS, G. J.
STRUCTURED DESIGN
DOC. DATE: MAY 1974
AUTHORS: STEVENS, W. P. MEYERS, G. J. CONSTANTINE, L. L.

MICHELL, S.
OBJECT-ORIENTED STRUCTURED DESIGN OF LAYERED PROTOCOL SYSTEMS
DOC. DATE: OCTOBER 1982
AUTHORS: BUHR, R. J. A. MICHELL, S.

MINE, H.
PERFORMANCE EVALUATION OF A FAULT-TOLERANT COMPUTING SYSTEM
DOC. DATE: JUNE 1979
AUTHORS: MINE, H. HATAYAMA, K.

MOR, S.
16-BIT BIPOLAR MICROPROCESSOR MARCHES TO STANDARD INSTRUCTION SET
DOC. DATE: APRIL 7, 1983
AUTHORS: MOR, S. HINGARH, H. VORA, M. ET AL

MORGAN, D.
A SURVEY OF METHODS FOR IMPROVING COMPUTER NETWORK RELIABILITY AND AVAILABILITY
DOC. DATE: NOVEMBER 1977
AUTHORS: MORGAN, D. TAYLOR, P. CUSTEAU, G.

MOSES, KURT
SIFT - AN ULTRA-RELIABLE AVIONIC COMPUTING SYSTEM
DOC. DATE: OCTOBER 1981
AUTHORS: MOSES, KURT

MOVAGHAR, A.
A BIBLIOGRAPHY ON FORMAL METHODS FOR SYSTEM SPECIFICATION, DESIGN AND VALIDATION
DOC. DATE: JAN 1982
AUTHORS: MEYER, J. F. FURCHTGOTT, D. MOVAGHAR, A.

MUELLER, FREDERICK R.
THE RACE BETWEEN C AND ADA MAY HAVE TWO WINNERS
DOC. DATE: APRIL 14, 1983
AUTHORS: MUELLER, FREDERICK R. TAFT, S. TUCKER

MURRAY, N.
WAVELENGTH DIVISION MULTIPLEXING FOR FUTURE SPACE STATION DATA SYSTEMS
DOC. DATE: AUGUST 25, 1983
AUTHORS: HENDRICKS, H. MURRAY, N.

MUSA, J. D.
SOFTWARE RELIABILITY MEASUREMENT - THE STATE OF THE ART
DOC. DATE: JUNE 14, 1982
AUTHORS: MUSA, J. D.

NELSON, H. G.
STATE-OF-THE-ART COMPUTER MONITORING EQUIPMENT
DOC. DATE: OCTOBER 1981
AUTHORS: NELSON, H. G.

NESS, W.
AUTOMATED RELIABILITY AND FAILURE EFFECTS METHODS FOR DIGITAL FLIGHT CONTROL AND AVIONICS SYSTEMS - VOLUME 1: EVALUATION
DOC. DATE: MARCH 1981
AUTHORS: NESS, W. MCCRARY. W. BRIDGEMAN, M.

NEUMANN, J.
CARE III PHASE III REPORT - TEST AND EVALUATION
DOC. DATE: NOVEMBER 1982
AUTHORS: STIFFLER, J. NEUMANN, J. BRYANT, L.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

NG, Y.
A UNIFIED RELIABILITY MODEL FOR FAULT-TOLERANT COMPUTERS
DOC. DATE: NOVEMBER 1980
AUTHORS: NG, Y. AVIZIENIS, A.

ORANEN, J.
SPECIFICATION AND ANALYSIS OF COMMUNICATION IN DISTRIBUTED DATA PROCESSING SYSTEMS
DOC. DATE: APRIL 1982
AUTHORS: ORANEN, J.

PETERSON, C.
TWO CHIPS ENDOW 32-BIT PROCESSOR WITH FAULT-TOLERANT ARCHITECTURE
DOC. DATE: APRIL 7, 1983
AUTHORS: PETERSON, C. DUZETTE, C. BUDDE, D. ET AL

PINKSTON, W.E.
SNEAK SOFTWARE ANALYSIS
DOC. DATE: 1983
AUTHORS: BURATTI, D.L. PINKSTON, W.E. SIMKINS, R.D.

POPEK, G. ET AL
LOCUS
DOC. DATE: DECEMBER 1, 1981
AUTHORS: POPEK, G. ET AL

PRADHAN, D. K.
A FAULT-TOLERANT COMMUNICATION ARCHITECTURE FOR DISTRIBUTED SYSTEMS
DOC. DATE: SEPTEMBER 1982
AUTHORS: PRADHAN, D. K. REDDY, S. M.

QUICK, M.
THE ANALYSIS OF THE STATISTICAL AND HISTORICAL INFORMATION GATHERED DURING THE DEVELOPMENT OF THE SHUTTLE ORBITER PRIMARY FLIGHT SOFTWARE
DOC. DATE: JUNE 1982
AUTHORS: SIMMONS, D. MARCHBANKS, M. QUICK, M.

RAMAMOORTHY, C. V.
DESIGN AND ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS
DOC. DATE: 1981
AUTHORS: RAMAMOORTHY, C. V. MA, Y. W.

(RAMAMOORTHY, C. V. CONTINUED)
ISSUES IN SOFTWARE RELIABILITY
DOC. DATE: JULY 1981
AUTHORS: RAMAMOORTHY, C. V. GANESH, S. L.

(RAMAMOORTHY, C. V. CONTINUED)
SOFTWARE RELIABILITY - STATUS AND PERSPECTIVES
DOC. DATE: JULY 1982
AUTHORS: RAMAMOORTHY, C. V. BASTANI, F. B.

RANDELL, B.
SYSTEM STRUCTURE FOR SOFTWARE FAULT TOLERANCE
DOC. DATE: JUNE 1975
AUTHORS: RANDELL, B.

(RANDELL, B. CONTINUED)
RELIABILITY AND INTEGRITY OF DISTRIBUTED COMPUTING SYSTEMS
DOC. DATE: JUNE 24, 1981
AUTHORS: RANDELL, B.

(RANDELL, B. CONTINUED)
THE NEWCASTLE CONNECTION OR UNIXES OF THE WORLD UNITE!
DOC. DATE: DECEMBER 1982
AUTHORS: BROWNBRIDGE, D. MARSHALL, L. RANDELL, B.

REDDY, S. M.
A FAULT-TOLERANT COMMUNICATION ARCHITECTURE FOR DISTRIBUTED SYSTEMS
DOC. DATE: SEPTEMBER 1982
AUTHORS: PRADHAN, D. K. REDDY, S. M.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

RHYNE, J. R.
DISTRIBUTED SYSTEM DESIGN DISCIPLINE: AN APPLICATION OF TELL
DOC. DATE: 1981
AUTHORS: RHYNE, J. R. DALPEZZO, R. G.

RITCHIE, D.
THE UNIX TIME-SHARING SYSTEM
DOC. DATE: 1974
AUTHORS: RITCHIE, D.

RITCHIE, DENNIS
THE C PROGRAMMING LANGUAGE
DOC. DATE: 1978
AUTHORS: KERNIGHAN, BRIAN RITCHIE, DENNIS

ROBINSON, A.
COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS
DOC. DATE: APRIL 1981
AUTHORS: HITT, E. F. BRIDGEMAN, M. ROBINSON, A.

ROBSON, D.
OBJECT-ORIENTED SOFTWARE SYSTEMS
DOC. DATE: AUGUST 1981
AUTHORS: ROBSON, D.

ROSS, D.
STRUCTURED ANALYSIS (SA): A LANGUAGE FOR COMMUNICATING IDEAS
DOC. DATE: JANUARY 1977
AUTHORS: ROSS, D.

RUDIN, H.
A VALIDATION TECHNIQUE FOR TIGHTLY COUPLED PROTOCOLS
DOC. DATE: JULY 1982
AUTHORS: RUDIN, H. WEST, C.

RUMMER, D. I.
MICROPROCESSOR SYSTEMS AND ARCHITECTURES FOR APPLICATIONS TO THE CONTROL AND PROTECTION OF ELECTRIC POWER SYSTEMS
DOC. DATE: 1979
AUTHORS: RUMMER, D. I. KEZUNOVIC, M.

SATKO, JAMES E.
IMPLEMENTATION LANGUAGES FOR DATA ABSTRACTIONS
DOC. DATE: MARCH 19, 1984
AUTHORS: LEACH, DANIEL M. SATKO, JAMES E.

SATYANAVAYANA, A.
A NEW ALGORITHM FOR THE RELIABILITY ANALYSIS OF MULTI-TERMINAL NETWORKS
DOC. DATE: OCTOBER 1981
AUTHORS: SATYANAVAYANA, A. HAGSTROM, J.

SCHWARTZ, R.
FORMAL SPECIFICATION AND MECHANICAL VERIFICATION OF SIFT: A FAULT-TOLERANT FLIGHT CONTROL SYSTEM
DOC. DATE: JULY 1982
AUTHORS: MELLIAR-SMITH, P. SCHWARTZ, R.

SCULL, J.
TECHNOLOGY NEEDS FOR AIR FORCE AUTONOMOUS SPACECRAFT
DOC. DATE: MARCH 16, 1982
AUTHORS: SCULL, J.

SEGALL, Z.
EVALUATION OF MULTIPROCESSOR INTERCONNECT STRUCTURES WITH THE CM* TESTBED
DOC. DATE: 1983
AUTHORS: WILSON, A. SIEWIOREK, DANIEL P. SEGALL, Z.

SEHR, R.
13 COMPANIES CLOSE IN ON LAN STANDARD PROPOSAL
DOC. DATE: MARCH 1983
AUTHORS: SEHR, R.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

SENN, E. H.
INTEGRATED VERIFICATION AND TESTING SYSTEM (IVTS) FOR HAL/S PROGRAMS
DOC. DATE: JULY 1983
AUTHORS: SENN, E. H. AMES, K. R. SMITH, K. A.

SEVEIK, K.
A SYSTEMATIC APPROACH TO THE PERFORMANCE MODELING OF COMPUTER SYSTEMS
DOC. DATE: 1979
AUTHORS: KEINZLE, M. SEVEIK, K.

SHAMAN, P.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL III - EVALUATION OF COMPUTER ARCHITECTURE VIA TEST PROGRAMS
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. SHAMAN, P.

SHINK, G.
A UNIFIED METHOD FOR EVALUATING REAL-TIME COMPUTER CONTROLLERS: A CASE STUDY
DOC. DATE: 1983
AUTHORS: SHINK, G. KRISHNA, C. LEE, Y.

SHOJA, G. C.
SOME EXPERIENCES OF IMPLEMENTING THE ADA CONCURRENCY FACILITIES ON A DISTRIBUTED MULTIPROCESSOR COMPUTER SYSTEM
DOC. DATE: OCTOBER 1982
AUTHORS: SHOJA, G. C. HALSALL, F. GRIMSDALE, R. L.

SIEWIOREK, DANIEL P.
RELIABILITY MODELS FOR MULTIPROCESSOR SYSTEMS WITH AND WITHOUT PERIODIC MAINTENANCE
DOC. DATE: JUNE 1977
AUTHORS: INGLE, A. SIEWIOREK, DANIEL P.

(SIEWIOREK, DANIEL P. CONTINUED)
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME VIII - CFA FINAL SELECTION
DOC. DATE: SEPTEMBER 1977
AUTHORS: CLEARWATERS, A. GORDON, R. SIEWIOREK, DANIEL P.

(SIEWIOREK, DANIEL P. CONTINUED)
THE MEASUREMENT AND ANALYSIS OF TRANSIENT ERRORS IN DIGITAL COMPUTER SYSTEMS
DOC. DATE: JUNE 1979
AUTHORS: MCCONNEL, S. R. SIEWIOREK, DANIEL P.

(SIEWIOREK, DANIEL P. CONTINUED)
A PERFORMANCE - RELIABILITY MODEL FOR COMPUTING SYSTEMS
DOC. DATE: OCTOBER 1980
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.

(SIEWIOREK, DANIEL P. CONTINUED)
IMPLEMENTATION AND PERFORMANCE EVALUATION OF COMPUTER FAMILIES
DOC. DATE: JUNE 1981
AUTHORS: SNOW, E. SIEWIOREK, DANIEL P.

(SIEWIOREK, DANIEL P. CONTINUED)
WORKLOAD, PERFORMANCE AND RELIABILITY OF DIGITAL COMPUTING SYSTEMS
DOC. DATE: JUNE 1981
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.

(SIEWIOREK, DANIEL P. CONTINUED)
THE THEORY AND PRACTICE OF RELIABLE SYSTEM DESIGN
DOC. DATE: 1982
AUTHORS: SIEWIOREK, DANIEL P. SWARZ, ROBERT S.

(SIEWIOREK, DANIEL P. CONTINUED)
A WORKLOAD DEPENDENT SOFTWARE RELIABILITY PREDICTION MODEL
DOC. DATE: JUNE 1982
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.

(SIEWIOREK, DANIEL P. CONTINUED)
EVALUATION OF MULTIPROCESSOR INTERCONNECT STRUCTURES WITH THE CM* TESTBED
DOC. DATE: 1983
AUTHORS: WILSON, A. SIEWIOREK, DANIEL P. SEGALL, Z.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

SIMKINS, R.D.
SNEAK SOFTWARE ANALYSIS
DOC. DATE: 1983
AUTHORS: BURATTI, D.L. PINKSTON, W.E. SIMKINS, R.D.

SIMMONS, D.
THE ANALYSIS OF THE STATISTICAL AND HISTORICAL INFORMATION GATHERED DURING THE DEVELOPMENT OF THE SHUTTLE ORBITER PRIMARY FLIGHT SOFTWARE
DOC. DATE: JUNE 1982
AUTHORS: SIMMONS, D. MARCHBANKS, M. QUICK, M.

SMITH, K. A.
INTEGRATED VERIFICATION AND TESTING SYSTEM (IVTS) FOR HAL/S PROGRAMS
DOC. DATE: JULY 1983
AUTHORS: SENN, E. H. AMES, K. R. SMITH, K. A.

SMITH, R.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL 1 - INTRODUCTION
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. COLEMAN, A. SMITH, R.

(SMITH, R. CONTINUED)
A FLEXIBLE DISTRIBUTED TESTBED FOR REAL-TIME APPLICATIONS
DOC. DATE: OCTOBER 1982
AUTHORS: MCDONALD, W. SMITH, R.

SMITH, T. BASIL, III
A DAMAGE- AND FAULT-TOLERANT INPUT/OUTPUT NETWORK
DOC. DATE: MAY 1975
AUTHORS: SMITH, T. BASIL, III

(SMITH, T. BASIL, III CONTINUED)
FTMP - A HIGHLY RELIABLE FAULT-TOLERANT MULTIPROCESSOR FOR AIRCRAFT
DOC. DATE: OCTOBER 1978
AUTHORS: HOPKINS, ALBERT L. SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(SMITH, T. BASIL, III CONTINUED)
DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER, VOLUME I, FTMP PRINCIPLES OF OPERATION
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(SMITH, T. BASIL, III CONTINUED)
DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER, VOLUME II, FTMP SOFTWARE
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(SMITH, T. BASIL, III CONTINUED)
DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER, VOLUME III, FTMP TEST AND EVALUATION
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.

(SMITH, T. BASIL, III CONTINUED)
FAULT TOLERANT PROCESSOR CONCEPTS AND OPERATION
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III

SNOW, E.
IMPLEMENTATION AND PERFORMANCE EVALUATION OF COMPUTER FAMILIES
DOC. DATE: JUNE 1981
AUTHORS: SNOW, E. SIEWIOREK, DANIEL P.

SOI, I.
RELIABILITY INDICES FOR TOPOLOGICAL DESIGN OF COMPUTER COMMUNICATION NETWORKS
DOC. DATE: DECEMBER 1981
AUTHORS: SOI, I. AGGARWAL, K.

SPRADLIN, R.
THE 757/767 FLIGHT MANAGEMENT SYSTEM LABORATORY TEST PROGRAM
AUTHORS: SPRADLIN, R.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

SPRADLIN, R. * CONTINUED *

(SPRADLIN, R. CONTINUED)
BOEING 757 AND 767 FLIGHT MANAGEMENT SYSTEM
DOC. DATE: NOVEMBER 20, 1980
AUTHORS: SPRADLIN, R.

SPRATT, E. B.
FURTHER DEVELOPMENTS ON THE CAMBRIDGE RING NETWORK AT THE UNIVERSITY OF KENT
DOC. DATE: 1982
AUTHORS: BINNS, S. E. DALLAS, I. N. SPRATT, E. B.

STAY, J. F.
HIPO AND INTEGRATED PROGRAM DESIGN
DOC. DATE: 1976
AUTHORS: STAY, J. F.

STERN, A.
STAGE - STATE RELIABILITY ANALYSIS TECHNIQUE
DOC. DATE: JUNE 1981
AUTHORS: STERN, A.

STERN, A. D.
INTEGRATED AIRFRAME/PROPULSION CONTROL SYSTEM ARCHITECTURES (IAPSA) STUDY
DOC. DATE: 1983
AUTHORS: STERN, A. D.

STEVENS, W. P.
STRUCTURED DESIGN
DOC. DATE: MAY 1974
AUTHORS: STEVENS, W. P. MEYERS, G. J. CONSTANTINE, L. L.

STIFFLER, J.
CARE III FINAL REPORT PHASE I VOLUME I
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.

(STIFFLER, J. CONTINUED)
CARE III FINAL REPORT PHASE I VOLUME II
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.

(STIFFLER, J. CONTINUED)
CARE III PHASE II REPORT USER'S MANUAL
DOC. DATE: SEPTEMBER 1982
AUTHORS: BRYANT, L. STIFFLER, J.

(STIFFLER, J. CONTINUED)
CARE III PHASE II REPORT - MATHEMATICAL DESCRIPTION
DOC. DATE: NOVEMBER 1982
AUTHORS: STIFFLER, J. BRYANT, L.

(STIFFLER, J. CONTINUED)
CARE III PHASE III REPORT - TEST AND EVALUATION
DOC. DATE: NOVEMBER 1982
AUTHORS: STIFFLER, J. NEUMANN, J. BRYANT, L.

STONE, H.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME II - SELECTION OF CANDIDATE ARCHITECTURES AND INITIAL SCREENING
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. STONE, H.

STREHLO, K.
INTEL TO BOLSTER 432 SALES WITH FAULT-TOLERANT COMPONENTS
DOC. DATE: MARCH 1983
AUTHORS: STREHLO, K.

SUTCLIFFE, P.
LESSONS LEARNED IN THE DEVELOPMENT OF THE 757/767 FLIGHT MANAGEMENT SYSTEM EQUIPMENT
AUTHORS: SUTCLIFFE, P.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

SVIRSKY, W.
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT - VOL VI - LIFE CYCLE COST MODELS
DOC. DATE: SEPTEMBER 1977
AUTHORS: COLEMAN, A. CORNYN, J. SVIRSKY, W.

SVOBODOVA, L.
COMPUTER PERFORMANCE MEASUREMENT AND EVALUATION METHODS
DOC. DATE: JUNE 1974
AUTHORS: SVOBODOVA, L.

(SVOBODOVA, L. CONTINUED)
RELIABILITY ISSUES IN DISTRIBUTED INFORMATION PROCESSING SYSTEM
DOC. DATE: JUNE 1979
AUTHORS: SVOBODOVA, L.

SWAMY, S.
A SYSTEMATIC APPROACH TO THE DESIGN OF FAULT TOLERANT DISTRIBUTED COMPUTER SYSTEMS
DOC. DATE: 1983
AUTHORS: COLEMAN, D. M. JORDAN, B. W., JR. SWAMY, S.

SWARZ, ROBERT S.
THE THEORY AND PRACTICE OF RELIABLE SYSTEM DESIGN
DOC. DATE: 1982
AUTHORS: SIEWIOREK, DANIEL P. SWARZ, ROBERT S.

SWERN, F.
MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: MCGOUGH, J. G. SWERN, F.

(SWERN, F. CONTINUED)
METHODOLOGY FOR MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR
DOC. DATE: OCTOBER 1981
AUTHORS: MCGOUGH, J. G. SWERN, F. BAVUSO, S. J.

(SWERN, F. CONTINUED)
MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR - PART II
DOC. DATE: 1983
AUTHORS: MCGOUGH, J. G. SWERN, F.

SWINGLE, W. J.
SPACE STATION INFORMATION SYSTEMS
DOC. DATE: 1983
AUTHORS: SWINGLE, W. J. MCKAY, C. W.

SYKES, D. J.
PERFORMANCE MODELING OF DISTRIBUTED SYSTEMS
AUTHORS: SYKES, D. J.

SYRBE, M.
THE DESCRIPTION OF FAULT-TOLERANT SYSTEMS
DOC. DATE: 1981
AUTHORS: SYRBE, M.

TAFT, S. TUCKER
THE RACE BETWEEN C AND ADA MAY HAVE TWO WINNERS
DOC. DATE: APRIL 14, 1983
AUTHORS: MUELLER, FREDERICK R. TAFT, S. TUCKER

TAKEUCHI, A.
OBJECT-ORIENTED DESCRIPTION ENVIRONMENT FOR COMPUTER HARDWARE
AUTHORS: TAKEUCHI, A.

TAYLOR, P.
A SURVEY OF METHODS FOR IMPROVING COMPUTER NETWORK RELIABILITY AND AVAILABILITY
DOC. DATE: NOVEMBER 1977
AUTHORS: MORGAN, D. TAYLOR, P. CUSTEAU, G.

TAYLOR, R.
REDUNDANT PROGRAMMING IN EUROPE
DOC. DATE: JANUARY 1981
AUTHORS: TAYLOR, R.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

TEICHROEW, D.
PSL/PSA: A COMPUTER AIDED TECHNIQUE FOR STRUCTURED DOCUMENTATION AND ANALYSIS OF INFORMATION PROCESSING
DOC. DATE: JANUARY 1977
AUTHORS: TEICHROEW, D. HERSHEY, E. A.

THEIS, D.
SPACECRAFT COMPUTERS: STATE-OF-THE-ART SURVEY
DOC. DATE: APRIL 1983
AUTHORS: THEIS, D.

TRIVEDI, K.
ANALYSIS OF M/G/2 - STANDBY REDUNDANT SYSTEM
AUTHORS: BACCELLI, F. TRIVEDI, K.

(TRIVEDI, K. CONTINUED)
DECOMPOSITION IN RELIABILITY ANALYSIS OF FAULT-TOLERANT SYSTEMS
AUTHORS: TRIVEDI, K. GEIST, R.

(TRIVEDI, K. CONTINUED)
DESIGN OF THE HYBRID AUTOMATED RELIABILITY PREDICTOR
AUTHORS: GEIST, R. TRIVEDI, K. DUGAN, J.

(TRIVEDI, K. CONTINUED)
THE INTEGRATION OF USER PERCEPTION IN THE HETEROGENEOUS M/M/2 QUEUE
AUTHORS: GEIST, R. TRIVEDI, K.

(TRIVEDI, K. CONTINUED)
A VALIDATION PROTOTYPE OF SYSTEM RELIABILITY IN LIFE-CRITICAL APPLICATIONS
DOC. DATE: 1980
AUTHORS: TRIVEDI, K. GAULT, J. CLARY, J.

(TRIVEDI, K. CONTINUED)
A MODEL FOR COMPUTER CONFIGURATION DESIGN
DOC. DATE: APRIL 1980
AUTHORS: TRIVEDI, K. KINICKI, R.

(TRIVEDI, K. CONTINUED)
VALIDATION METHODS RESEARCH FOR FAULT-TOLERANT AVIONICS AND CONTROL SYSTEM SUB-WORKING GROUP MEETING - CARE III PEER REVIEW
DOC. DATE: SEPTEMBER 1980
AUTHORS: TRIVEDI, K. CLARY, J.

(TRIVEDI, K. CONTINUED)
PROBABILITY AND STATISTICS WITH RELIABILITY, QUEUING AND COMPUTER SCIENCE APPLICATIONS
DOC. DATE: 1982
AUTHORS: TRIVEDI, K.

(TRIVEDI, K. CONTINUED)
OPTIMAL DESIGN OF MULTILEVEL STORAGE HIERARCHIES
DOC. DATE: MARCH 1982
AUTHORS: GEIST, R. TRIVEDI, K.

(TRIVEDI, K. CONTINUED)
ANALYTIC QUEUEING THEORY FOR PROGRAMS WITH INTERNAL CONCURRENCY
DOC. DATE: 1983
AUTHORS: HEIDELBERGER, P. TRIVEDI, K.

(TRIVEDI, K. CONTINUED)
ULTRA-HIGH RELIABILITY PREDICTION FOR FAULT-TOLERANT COMPUTER SYSTEMS
DOC. DATE: FEBRUARY 1983
AUTHORS: GEIST, R. TRIVEDI, K.

(TRIVEDI, K. CONTINUED)
TASK ALLOCATION IN FAULT-TOLERANT DISTRIBUTED SYSTEMS
DOC. DATE: MARCH 1983
AUTHORS: BANNISTER, J. TRIVEDI, K.

TSAY, D. P.
DESIGN OF A NETWORK OPERATING SYSTEM FOR THE DISTRIBUTED DOUBLE-LOOP COMPUTER NETWORK (DDLCN)
DOC. DATE: 1982
AUTHORS: LIU, M. T. TSAY, D. P. LIAN, R. C.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

UCZEKAJ, J.S.
IMPLEMENTATION OF DO-178 IN COMMERCIAL AIRBORNE SOFTWARE DESIGN
DOC. DATE: 1983
AUTHORS: UCZEKAJ, J.S.

VAN RUMSTE, M.
THE IAPX432, A NEXT GENERATION MICROPROCESSOR
DOC. DATE: 1983
AUTHORS: VAN RUMSTE, M.

VOGES, U.
SOFTWARE DIVERSITY IN REACTOR PROTECTION SYSTEMS: AN EXPERIMENT
DOC. DATE: 1979
AUTHORS: GMEINER, L. VOGES, U.

VORA, M. ET AL
16-BIT BIPOLAR MICROPROCESSOR MARCHES TO STANDARD INSTRUCTION SET
DOC. DATE: APRIL 7, 1983
AUTHORS: MOR, S. HINGARH, H. VORA, M. ET AL

WAGNER, J.
COMPUTER FAMILY ARCH SEL COMM-FINAL RPT-V5-PROC FOR AND RESULTS OF THEEVAL UATION OF SOFTWARE BASES OF THE CANDIDATE ARCH FOR THE MIL COMPUTER FAMILY
DOC. DATE: SEPTEMBER 1977
AUTHORS: LIEBLEIN, E. WAGNER, J. CLEARWATERS, A.

WARD, A. O.
A CONSISTENT APPROACH TO THE DEVELOPMENT OF SYSTEM REQUIREMENTS AND SOFTWARE DESIGN
DOC. DATE: OCTOBER 1981
AUTHORS: WARD, A. O.

WEIMANN, A.
FUNCTIONAL VERSUS COMMUNICATION STRUCTURES IN MODERN AVIONIC SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: BRAMMER, K. WEIMANN, A.

WEST, C.
A VALIDATION TECHNIQUE FOR TIGHTLY COUPLED PROTOCOLS
DOC. DATE: JULY 1982
AUTHORS: RUDIN, H. WEST, C.

WILSON, A.
EVALUATION OF MULTIPROCESSOR INTERCONNECT STRUCTURES WITH THE CM* TESTBED
DOC. DATE: 1983
AUTHORS: WILSON, A. SIEWIOREK, DANIEL P. SEGALL, Z.

WOLF, J. K.
GENERALIZED POLLING ALGORITHMS FOR DISTRIBUTED SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: WOLF, J. K.

WOODS ET AL
HMOS PROCESSING IMPROVES RADIATION RESISTANCE OF VLSI MEMORIES AND MICROPROCESSORS
DOC. DATE: NOVEMBER 30, 1982
AUTHORS: DAVIS WOODS ET AL

WU, L.
PERFORMABILITY EVALUATION OF THE SIFT COMPUTER
DOC. DATE: JUNE 1980
AUTHORS: MEYER, J. F. FURCHTGOTT, D. WU, L.

(WU, L. CONTINUED)
MODELS FOR EVALUATING THE PERFORMABILITY OF DEGRADABLE COMPUTING SYSTEMS
DOC. DATE: JUNE 1982
AUTHORS: WU, L.

WULF, WILLIAM A.
TRENDS IN THE DESIGN AND IMPLEMENTATION OF PROGRAMMING LANGUAGES
DOC. DATE: JANUARY 1980
AUTHORS: WULF, WILLIAM A.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY AUTHORS, THEN BY DOCUMENT DATE

ZEMPOLICH, B. A.
A TUTORIAL ON DISTRIBUTED PROCESSING IN AIRCRAFT/AVIONICS APPLICATIONS
DOC. DATE: OCTOBER 1981
AUTHORS: ZEMPOLICH, B. A.

(ZEMPOLICH, B. A. CONTINUED)
ECONOMIC CONSIDERATIONS FOR REAL-TIME NAVAL AIRCRAFT/AVIONIC DISTRIBUTED COMPUTER CONTROL SYSTEMS
DOC. DATE: OCTOBER 1981
AUTHORS: ZEMPOLICH, B. A.

ZUCKERMAN, SUSAN LANA
PROBLEMS WITH THE MULTITASKING FACILITIES IN THE ADA PROGRAMMING LANGUAGE
DOC. DATE: MAY 1981
AUTHORS: ZUCKERMAN, SUSAN LANA

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

A BIBLIOGRAPHY ON FORMAL METHODS FOR SYSTEM SPECIFICATION, DESIGN AND VALIDATION

DOC. TYPE: REPORT
DOC. DATE: JAN 1982
AUTHORS: MEYER, J. F. FURCHTGOTT, D. MOVAGHAR, A.
SOURCE, PUB. BY, ETC: THE UNIVERSITY OF MICHIGAN
DOCUMENT NUMBER: SYSTEMS ENGINEERING LAB REPORT NO. 162
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS REPORT CONTAINS THE RESULTS OF A LITERATURE SURVEY DESIGNED TO CLASSIFY CURRENT LITERATURE ON FORMAL METHODS THAT MIGHT BE MEANINGFULLY EXPLOITED IN THE SPECIFICATION, DESIGN, AND VALIDATION OF AVIONIC SYSTEMS (WHERE VALIDATING INCLUDES VERIFICATION, TESTING AND EVALUATION). THE SPECIFIC LITERATURE SEARCHED INCLUDES JOURNAL PAPERS, CONFERENCE PAPERS AND TECHNICAL REPORTS PUBLISHED DURING THE LAST FIVE YEARS (BEGINNING IN 1977). THE ARTICLES ARE CLASSIFIED ACCORDING TO FIVE TOPIC AREAS: SPECIFICATION, DESIGN, VERIFICATION, TESTING AND EVALUATION. BECAUSE THE SURVEY WAS COMPLETED IN SEPTEMBER, 1981, NO CITATIONS APPEARING AFTER THAT DATE ARE INCLUDED IN THE RESULTING BIBLIOGRAPHY.

A COMPARISON OF THE PROGRAMMING LANGUAGES C AND PASCAL

DOC. TYPE: ARTICLE
DOC. DATE: MARCH 1982
AUTHORS: FEUER, ALAN R. GEHANI, NARAIN H.
SOURCE, PUB. BY, ETC: COMPUTING SURVEYS, VOL 14. NO. 1 MAR 1982, ACM
FILED BY (NAME): KNOSP, A. A.
ABSTRACT:
THE LANGUGES C AND PASCAL ARE GROWING IN POPULARITY, PARTICULARLY AMONG PROGRAMMERS OF SMALL COMPUTERS. IN THIS PAPER WE SUMMARIZE AND COMPARE THE TWO LANGUAGES COVERING THEIR DESIGN PHILOSOPHIES, THEIR HANDLING OF DATA TYPES, THE PROGRAMMING FACILITIES THEY PROVIDE, THE IMPACT OF THESE FACILITIES ON THE QUALITY OF PROGRAMS, AND HOW USEFUL THE FACILITIES ARE FOR PROGRAMMING IN A VARIETY OF APPLICATION DOMAINS.

A CONSISTENT APPROACH TO THE DEVELOPMENT OF SYSTEM REQUIREMENTS AND SOFTWARE DESIGN

DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: WARD, A. D.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): SZULEWSKI, P.
ABSTRACT:
TIC #AD-A109-274. THE AUTHOR IS WITH BRITISH AEROSPACE PUBLIC LIMITED COMPANY, AIRCRAFT GROUP, WARTON DIVISION, PRESTON PR4 1AX, UNITED KINGDOM.

A DAMAGE- AND FAULT-TOLERANT INPUT/OUTPUT NETWORK

DOC. TYPE: ARTICLE
DOC. DATE: MAY 1975
AUTHORS: SMITH, T. BASIL, III
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, MAY 1975, PP 505-512
FILED BY (NAME): LALA, J. H.
ABSTRACT:
A DAMAGE- AND FAULT-TOLERANT INPUT/OUTPUT (I/O) NETWORK IS PRESENTED AS AN ALTERNATIVE TO I/O BUSES. SUCH A NETWORK DIFFERS MOST SIGNIFICANTLY FROM TELECOMMUNICATIONS NETWORKS (SUCH AS THE ARPA NET) IN THAT THE NODES OF THIS NET ARE VERY SIMPLE AND DO NOT PERFORM SUCH TASKS AS MESSAGE OR PACKET SWITCHING AND BUFFERING. DATA ROUTING THROUGH THE NET IS DIRECTLY CONTROLLED BY A CENTRALIZED PROCESSOR WITH CONTROL ALGORITHMS BEING IMPLEMENTED IN SOFTWARE. DESCRIPTIONS OF BOTH THE NETWORK HARDWARE AND THE MOST VITAL POINTS OF THE CONTROL SOFTWARE ARE PROVIDED. IT IS BELIEVED THAT THIS NETWORK CONSTRUCTION REPRESENTS BOTH AN ECONOMICALLY VIABLE ALTERNATIVE TO DATA BUSES AND A SIGNIFICANT IMPROVEMENT IN RELIABILITY AND SURVIVABILITY, PARTICULARLY IN THOSE APPLICATIONS WHERE PHYSICAL DAMAGE TO ELEMENTS OF THE I/O SYSTEM IS A POSSIBILITY, SUCH AS IN AIRCRAFT, SHIPBOARD, RAPID TRANSIT, AND INDUSTRIAL SITUATIONS.

A FAULT-TOLERANT COMMUNICATION ARCHITECTURE FOR DISTRIBUTED SYSTEMS

DOC. TYPE: ARTICLE
DOC. DATE: SEPTEMBER 1982
AUTHORS: PRADHAN, D. K. REDDY, S. M.
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-31, #9, 863-870
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
A COMMUNICATION ARCHITECTURE FOR DISTRIBUTED PROCESSORS IS PRESENTED HERE. THIS ARCHITECTURE IS BASED ON A NEW TOPOLOGY WE HAVE DEVELOPED, ONE WHICH INTERCONNECTS N NODES BY USING RN LINKS WHERE THE MAXIMUM INTERNODE DISTANCE IS LOG, N, AND WHERE EACH NODE HAS, AT MOST, 2R, I/O PORTS. IT IS ALSO SHOWN THAT THIS NETWORK IS FAULT-TOLERANT, BEING ABLE TO TOLERATE UP TO (R-1) NODE FAILURES. ONE OF THE PARTICULARLY ATTRACTIVE FEATURES OF THIS NETWORK IS THAT IT ALLOWS FOR SIMPLE ROUTING AS WELL AS FOR EASY DISTRIBUTED FAULT-DIAGNOSIS. ALGORITHMS ARE ALSO DEVELOPED HERE FOR THE PURPOSE OF ROUTING MESSAGES FRON NODE TO NODE; THESE ARE USEFUL BOTH WITH AND WITHOUT THE PRESENCE OF FAULTS IN THE NETWORK. A PROCEDURE IS DEVELOPED, TOO, WHEREBY EACH FAULT-FREE NODECAN DIAGNOSE THE FAULT

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

A FAULT-TOLERANT COMMUNICATION ARCHITECTURE FOR DISTRIBUTED SYSTEMS * CONTINUED *
NODES INDEPENDENTLY, WITHOUT THE USE OF ANY CENTRAL OBSERVER.

A FLEXIBLE DISTRIBUTED TESTBED FOR REAL-TIME APPLICATIONS
DOC. TYPE: ARTICLE
DOC. DATE: OCTOBER 1982
AUTHORS: MCDONALD, W. SMITH, R.
SOURCE, PUB. BY, ETC: COMPUTER
DOCUMENT NUMBER: 0018-9162/82/1000-0025
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS ARTICLE DESCRIBES A FLEXIBLE DISTRIBUTED TESTBED THAT IS BEING DEVELOPED TO SUPPORT THE DEVELOPMENT, ANALYSIS, TEST, EVALUATION, AND VALIDATION OF RESEARCH IN DISTRIBUTED COMPUTING FOR REAL-TIME APPLICATIONS. THE TESTBED NOT ONLY PROVIDES THE RESOURCES FOR EXPERIMENTALLY OBTAINING QUANTITATIVE RESULTS, BUT ALSO SERVES AS A FOCAL POINT FOR THE RESEARCH, INTEGRATING RELATED RESEARCH ACTIVITIES AND PROVIDING A MECHANISM FOR TECHNOLOGY TRANSFER TO ASSOCIATED RESEARCH EFFORTS.

A LOCAL-AREA COMMUNICATION NETWORK BASED ON A RELIABLE TOKEN-RING SYSTEM
DOC. TYPE: PAPER
DOC. DATE: 1982
AUTHORS: BUX, W. CLOSS, F. JANSON, P. A., ET AL
SOURCE, PUB. BY, ETC: PROC IFIP TC 6 INTL SYMP ON LOCAL CMPTR NTWRKS
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE ARCHITECTURE OF A LOCAL-AREA COMMUNICATION NETWORK DESIGNED AND IMPLEMENTED AT THE IBM ZURICH RESEARCH LABORATORY IS DESCRIBED. A TOKEN-RING SUBNETWORK ALLOWS FOR A LOW-COST ENTRY POINT; GROWTH POTENTIAL TO LARGE NETWORKS IS PROVIDED THROUGH INTERCONNECTING RINGS BY BRIDGES AND A HIGH-PERFORMANCE STORE-AND-FORWARD NODE, CALLED BLOCK SWITCH. TO ENSURE RELIABLE TOKEN-RING OPERATION, WE EMPLOY THE CONCEPT OF A MONITOR FUNCTION. WE ALSO SHOW HOW, IN ADDITION TO THE TRANSMISSION OF DATA FRAMES, A TOKEN-RING CAN PROVIDE SYNCHRONOUS CHANNELS, E.G., FOR VOICE SERVICES. ALL FUNCTIONS ABOVE NETWORK ACCESS AND THE BASIC TRANSPORT MECHANISM ARE EXECUTED ON AN END-TO-END BASIS. THUS, BLOCK SWITCH AND BRIDGES CAN BE COMPLETELY IMPLEMENTED IN HARDWARE, ALLOWING EASY OBTAINMENT OF AN AGGREGATE THROUGHPUT CAPACITY IN THE ORDER OF 100 MBPS.

AUTHORS: W. BUX, F. CLOSS, P. A. JANSON, K. KUMMERLE, H. R. MILLER, AND E. ROTHAUSER. ALL ARE WITH THE IBM ZURICH RESEARCH LABORATORY, 8803 RUSCHLIKON, SWITZERLAND.

THE IFIP TC 6 INTERNATIONAL IN-DEPTH SYMPOSIUM ON LOCAL COMPUTER NETWORKS WAS HELD AT FLORENCE, ITALY, 19-21 APRIL, 1982. PROC'GS PUB'D BY N. HOLLAND PUB'G CO., AMSTERDAM, NEW YORK, OXFORD.

A METHODOLOGY FOR DESIGN OF DIGITAL SYSTEMS - SUPPORTED BY SARA AT THE AGE OF ONE
DOC. TYPE: PAPER
DOC. DATE: 1978
AUTHORS: ESTRIN, G.
SOURCE, PUB. BY, ETC: NATIONAL COMPUTER CONFERENCE
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
ONE HOPE IS THAT THERE WILL EVOLVE EFFECTIVE METHODS FOR COMPOSING SUCH SYSTS FROM DEFINED, WELL BEHAVED BUILDING BLOCKS WHOSE COMPOSITE BEHAVIOR CAN BE SHOWN TO MEET PRESTATED REQ. A SECOND HOPE IS THAT STARTING FROM WELL FORMULATED REQ AND AN INITIAL ABSTRACT SOLUTION SYS, THERE WILL EVOLVE HELPFUL GUIDELINES FOR STRUCTURAL PARTITION AND EFFECTIVE ALGORITHMS FOR BEHAVIORAL REFINEMENT. THE REFINEMENT PROC SHOULD CONSERVE DESIRABLE PROPERTIES THROUGH AS MANY LEVELS OF ABSTRACTION AS THE DESIGN NEEDS. A THIRD HOPE IS THAT A NEW DIMENSION FOR ARCHITECTURE OF COMPUTER SYSTS WILL EMERGE FROM THESE DESIGN METHODS AND PERMIT US TO EFFECTIVELY USE THE OUTPOURING OF LARGE SCALE INTEG DEVICES.

REALIZATION OF THESE HOPES AND THEIR JOINT USE ARE THE GOAL OF THE METHODOLOGY DISCUSSED IN THE FIRST PART OF THIS PAPER. THE SECOND PART DESCRIBES THE STATE OF THE SYS IMPLEMENTED TO SUPPORT A DESIGNER ATEMPTING TO USE THE METHODOLOGY. A COMPANION PAPER ILLUSTRATES APPLICATION OF OUR METHODS TO DESIGN OF SOFTWARE.

THE SYS ARCHITECTS APPRENTICE, SARA, IS ONLY 1 YEAR OLD. THE ROOTS OF THIS WORK WERE REVIEWED IN SHORT BY THE AUTHOR IN ONE OF A SET OF PRESENTATIONS WHICH ANNOUNCED SARA EARLY IN 1977.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

A MODEL FOR COMPUTER CONFIGURATION DESIGN

DOC. TYPE: ARTICLE
DOC. DATE: APRIL 1980
AUTHORS: TRIVEDI, K. KINICKI, R.
SOURCE, PUB. BY, ETC: COMPUTER
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
COMPUTER SYSTEM PERFORMANCE DEPENDS UPON INTERACTIONS AMONG THE SOFTWARE COMPONENTS, THE HARDWARE COMPONENTS, AND THE USER WORKLOAD. UNDERSTANDING THE NATURE OF THESE INTERDEPENDENCES REQUIRES COLLECTING EXPERIMENTAL DATA AND DEVELOPING A MECHANISM FOR ANALYZING SYSTEM BEHAVIOR AS A FUNCTION OF COMPONENT VARIATIONS. RECENTLY, ANALYTIC MODELS HAVE EMERGED AS A VIABLE MEANS OF UNRAVELING THE COMPLEXITY OF MODERN CONFIGURATIONS INTO A MATHEMATICALLY TRACTABLE FORM. ACCEPTED ANALYTIC METHODS FOR PERFORMANCE ANALYSIS, GIVEN THE WORKLOAD PARAMETERS AND THE SYSTEM CONFIGURATION, HAVE BEEN ESTABLISHED. HOWEVER, SYSTEMATIC METHODS FOR SYSTEM SYNTHESIS WHICH ENABLE SYSTEM PARAMETERS TO BE DERIVED FROM DESIGN REQUIREMENTS ARE NOT WELL DEVELOPED. THIS ARTICLE PRESENTS AN OPTIMIZATION MODEL FOR DETERMINING DEVICE SPEEDS AND MAIN MEMORY SIZE WHICH WILL MAXIMIZE SYSTEM THROUGHPUT WITHIN A FIXED BUDGET. THE COMPUTER SYSTEM IS MODELED AS A CLOSED QUEUEING NETWORK OF THE BCMP CLASS WITH A SINGLE JOB TYPE AND A SINGLE SERVER AT EACH NODE IN THE NETWORK. EXPERIMENTS HAVE SHOWN THAT SUCH QUEUEING MODELS PREDICT PERFORMANCE MEASURES SUCH AS DEVICE UTILIZATION AND SYSTEM THROUGHPUT TO WITHIN 5 PERCENT.

A NEW ALGORITHM FOR THE RELIABILITY ANALYSIS OF MULTI-TERMINAL NETWORKS

DOC. TYPE: ARTICLE
DOC. DATE: OCTOBER 1981
AUTHORS: SATYANAVAYANA, A. HAGSTROM, J.
SOURCE, PUB. BY, ETC: IEEE TRANS ON RELIABILITY, VOL R-30, NO.4
DOCUMENT NUMBER: 0018-9529/81/1000-0325
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IN A PROBABILISTIC NETWORK, SOURCE-TO-MULTIPLE-TERMINAL RELIABILITY (SMT RELIABILITY) IS THE PROBABILITY THAT A SPECIFIED VERTEX CAN REACH EVERY OTHER VERTEX. THIS PAPER DERIVES A NEW TOPOLOGICAL FORMULA FOR THE SMT RELIABILITY OF PROBABILISTIC NETWORKS. THE FORMULA GENERATES ONLY NON-CANCELLING TERMS. THE NON-CANCELLING TERMS IN THE RELIABILITY EXPRESSION CORRESPOND ONE-TO-ONE WITH THE ACYCLIC T-SUBGRAPHS OF THE NETWORK. AN ACYCLIC T-SUBGRAPH IS AN ACYCLIC GRAPH IN WHICH EVERY LINK IS IN AT LEAST ONE SPANNING ROOTED TREE OF THE GRAPH. THE SIGN TO BE ASSOCIATED WITH EACH TERM IS EASILY COMPUTED BY COUNTING THE VERTICES AND LINKS IN THE CORRESPONDING SUBGRAPH.

OVERALL RELIABILITY IS THE PROBABILITY THAT EVERY VERTEX CAN REACH EVERY OTHER VERTEX IN THE NETWORK. FOR AN UNDIRECTED NETWORK, IT IS SHOWN THE SMT RELIABILITY IS EQUAL TO THE OVERALL RELIABILITY. THE FORMULA IS GENERAL AND APPLIES TO NETWORKS CONTAINING DIRECTED OR UNDIRECTED LINKS. FURTHERMORE LINK FAILURES IN THE NETWORK CAN BE S-DEPENDENT.

AN ALGORITHM IS PRESENTED FOR GENERATING ALL ACYCLIC T-SUBGRAPHS AND COMPUTING THE RELIABILITY OF THE NETWORK. THE RELIABILITY EXPRESSION IS OBTAINED IN SYMBOLIC FACTORED FORM.

A NEW APPROACH FOR NETWORK RELIABILITY ANALYSIS

DOC. TYPE: ARTICLE
DOC. DATE: OCTOBER 1982
AUTHORS: DEVERMEYER, B.
SOURCE, PUB. BY, ETC: IEEE TRANS ON RELIABILITY, VOL R-31, NO.4
DOCUMENT NUMBER: 0018-9529/82/1000-0350
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS PAPER PRESENTS A NOVEL APPROACH TO NETWORK RELIABILITY ANALYSIS BASED ON A NETWORK FUNCTION. ONCE THIS FUNCTION HAS BEEN CALCULATED FOR A GIVEN SYSTEM, THE NETWORK IS NO LONGER NEEDED IN THE ANALYSIS. THE APPROACH IS FAR MORE EFFICIENT THAN MONTE CARLO SIMULATION AND MUCH MORE FLEXIBLE THAN CUT-SET TECHNIQUES.

A PERFORMANCE - RELIABILITY MODEL FOR COMPUTING SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1980
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.
SOURCE, PUB. BY, ETC: PROC. 1980 INT SYMP, FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IN THIS PAPER SOME MEASURES ARE PRESENTED THAT CHARACTERIZE BOTH THE PERFORMANCE AND RELIABILITY OF DIGITAL COMPUTING SYSTEMS IN TIME SHARING ENVIRONMENTS FROM A USER VIEWPOINT. THE MEASURES (APPARENT CAPACITY AND EXPECTED ELAPSED TIME REQUIRED TO CORRECTLY EXECUTE A GIVEN PROGRAM) ARE BASED ON A MATHEMATICAL MODEL BUILT UPON TRADITIONAL ASSUMPTIONS. THE MODEL IS A HYBRID IN THAT IT USES STATISTICS GATHERED FROM A REAL SYSTEM WHILE GIVING ANALYTICAL EXPRESSIONS FOR OTHER STATISTICS SUCH AS THE EXPECTED ELAPSED TIME. THE MAIN PARAMETERS OF THE MODEL ARE THE SYSTEM WORKLOAD AND THE DISTRIBUTION OF THE TIME BETWEEN ERRORS. ALTHOUGH STILL LIMITED BECAUSE OF THE RESTRICTIVE ASSUMPTIONS USED,

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

A PERFORMANCE - RELIABILITY MODEL FOR COMPUTING SYSTEMS * CONTINUED *

THE MODEL GIVES QUANTITATIVE RESULTS ABOUT HOW MUCH A USER CAN EXPECT FROM A TIME SHARING SYSTEM, AS A FUNCTION OF THE SYSTEM WORKLOAD AND RELIABILITY. FOR EXAMPLE, THIS STUDY MEASURED A FOUR TO ONE RANGE IN MEAN TIME TO SYSTEM FAILURE AS A FUNCTION OF SYSTEM LOAD. FOR THE MAXIMUM LOAD PERIOD MEASURED THE MODEL PREDICTS A 40% CONTRIBUTION FROM SYSTEM UNRELIABILITY TO EXPECTED COMPUTATION TIME FOR A PROGRAM THAT COULD REQUIRE 30 MINUTES OF CPU TIME IN AN UNLOADED SITUATION.

A PROPOSAL TO APPLY ADA AND OBJECT ARCHITECTURE TO SATISFY THE DATA REQUIREMENTS OF THE NASA SPACE STATION

DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: BOWN, R. L. MCKAY, C. W.
SOURCE, PUB. BY, ETC: AIAA COMPUTERS IN AEROSPACE IV CONF
DOCUMENT NUMBER: AIAA-83-2332-CP (NOT INCL IN CONF PROC'GS)
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:

THE PROPOSED NASA SPACE STATION MAY BE UNDER ACTIVE DEVELOPMENT DURING THE MID TO LATE 1980'S. THE DEVELOPMENT PHASE OF THE DATA MANAGEMENT SYSTEM CAN BENEFIT FROM THE USE OF MODERN COMPUTER SYSTEM TECHNOLOGIES. THE ADA COMPUTER PROGRAMMING LANGUAGE AND THE OBJECT ORIENTED ARCHITECTURE OF THE INTEL IAPX-432 MICRO-MAINFRAME COMPUTER SUSTEM FORM AN AVAILABLE MODERN COMPLEMENTARY SOFTWARE/HARDWARE TOOL SET. THIS PAPER EXAMINES THE STRUCTURES OF THE SPACE STATION DATA MANAGEMENT SYSTEM ANDITS POSSIBLE USE OF THE ADA/432 TOOL SET. SEVERAL PROOF OF PRINCIPLE DEVELOPMENT PROJECTS ARE RELATED TO INTERFACE STANDARDS, SUBSYSTEM DEVELOPMENT, AND DISTRIBUTED RELATIONAL DATA BASE MANAGEMENT.

THE AUTHORS ARE WITH THE UNIVERSITY OF HOUSTON - CLEAR LAKE, HOUSTON, TX.

A RELIABILITY AND LIFE-CYCLE EVALUATION TOOL FOR FAULT-TOLERANT SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: JUNE 1, 1981
AUTHORS: MAKAM, S. AVIZIENIS, A.
SOURCE, PUB. BY, ETC: 12TH ANN INTL SYMP ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:

THIS PAPER DESCRIBES A PROGRAM CALLED ARIES 81 (AUTOMATED RELIABILITY INTERACTIVE ESTIMATION SYSTEM) WHICH IS DESIGNED TO PROVIDE STUDENTS, RESEARCHERS, AND DESIGNERS WITH AN ENVIRONMENT IN WHICH THEY MAY DEFINE AND ANALYZE THE RELIABILITY MODELS FOR FAULT-TOLERANT COMPUTERS. ARIES 81 IS AN EXTENDED AND IMPROVED VERSION OF A RELIABILITY ESTIMATION PROGRAM CALLED ARIES 76. THE USER OF ARIES 81 IS PRESENTED WITH A MATHEMATICAL FRAMEWORK OF ANALYSIS -- STOCHASTIC HOMOGENEOUS MARKOV PROCESS, IN WHICH TO MODEL THE SYSTEM HE IS DESIGNING. ARIES 81 HAS THE CAPABILITY TO CONSTRUCT, PROCESS, AND EVALUATE THE PREDEFINED SPECIFIC MODELS FOR CLOSED, REPAIRABLE, AND PERIODICALLY RENEWED CLOSED FAULT-TOLERANT SYSTEMS. IT CAN ALSO PROCESS AND EVALUATE ANY NEWLY DEFINED AND USER CONSTRUCTED MODELS FOR MORE COMPLEX SYSTEMS AND CAN MODEL BOTH PERMANENT AND TRANSIENT FAULT RECOVERY. BOTH RELIABILITY AND LIFE-CYCLE ANALYSIS CAN BE PERFORMED WITH THE AID OF ARIES 81.

A REQUIREMENTS ENGINEERING METHODOLOGY FOR REAL-TIME PROCESSING REQUIREMENTS

DOC. TYPE: ARTICLE
DOC. DATE: 1977
AUTHORS: ALFORD, M. W.
SOURCE, PUB. BY, ETC: IEEE TR ON SFTWR ENGRG, V SE-3, #1, PP 60-69
FILED BY (NAME): SZULEWSKI, P.

A SIMULATOR FOR RELIABILITY PREDICTIONS OF FAULT-TOLERANT SYSTEM ARCHITECTURES

DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1982
AUTHORS: MARINOS, P.
SOURCE, PUB. BY, ETC: NAVAL RESEARCH LAB
DOCUMENT NUMBER: NRL-MR-4934
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:

THIS STUDY IS CONCERNED WITH A SIMULATION TECHNIQUE SUITABLE FOR PREDICTING THE RELIABILITY OF FAULT-TOLERANT SYSTEM ARCHITECTURES.

GENERAL FAULT-OCCURRENCE PROBABILITY DENSITY FUNCTIONS ARE INTRODUCED, AND TECHNIQUES FOR GENERATING POSTULATED FAULT ENVIRONMENTS USING THESE FUNCTIONS ARE PRESENTED. METHODS FOR ASSIGNING FAULTS TO SUBSYSTEMS OF A SYSTEM EXPOSED TO A GIVEN FAULT-ENVIRONMENT ARE DISCUSSED AND THE UTILITY OF THESE METHODS FOR MAKING RELIABILITY PREDICTIONS IS ILLUSTRATED WITH A SPECIFIC EXAMPLE DERIVED FROM THE RADAR FIELD.

THE SIMULATOR WHICH HAS BEEN WRITTEN IN FORTRAN-77 IS HIGHLY INTERACTIVE AND USER ORIENTED, AND PROVIDES IN ADDITION TO SYSTEM RELIABILITY ESTIMATES, SYSTEM MTBF AND AVAILABILITY ESTIMATES UNDER VARIOUS REPAIR STRATEGIES AND DEGREES OF FAULT-COVERAGE.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

A SPECIFICATION-ORIENTED MULTI-VERSION SOFTWARE EXPERIMENT
DOC. TYPE: PREPRINT
DOC. DATE: 1983
AUTHORS: KELLY, JOHN P. J. AVIZIENIS, ALGIRDAS
SOURCE, PUB. BY, ETC: 13TH ANN INTL SYMP ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): FURTEK, F. C.

A SURVEY OF METHODS FOR IMPROVING COMPUTER NETWORK RELIABILITY AND AVAILABILITY
DOC. TYPE: ARTICLE
DOC. DATE: NOVEMBER 1977
AUTHORS: MORGAN, D. TAYLOR, P. CUSTEAU, G.
SOURCE, PUB. BY, ETC: COMPUTER
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
AS COMPUTER NETWORKS ARE INCREASINGLY PUT INTO USE, THE ORGANIZATIONS THEY SERVE ARE BECOMING INCREASINGLY CONCERNED ABOUT DATA AND NETWORK RELIABILITY AND AVAILABITITY. UNTIL RECENTLY, LITTLE EMPHASIS WAS PLACED ON MAKING COMPUTER SOFTWARE RELIABLE (EXCEPT IN MILITARY AND COMMUNICATIONS APPLICATIONS), AND LITTLE CONSIDERATION WAS GIVEN TO ENHANCING COMPUTER NETWORK RELIABILITY AND AVAILABILITY.

IN THE INTEREST OF FOCUSING ATTENTION ON THE SUBJECT, THIS PAPER SURVEYS THE PROBLEMS THAT ADVERSELY AFFECT COMPUTER NETWORK RELIABILITY AND AVAILABILITY, SURVEYS METHODS OF PREVENTING AND COPING WITH SUCH PROBLEMS, AND POINTS OUT AREAS NEEDING FURTHER STUDY. THE INTRODUCTORY SECTION OF THE PAPER DEFINES SOME TERMS AND EXPLAINS SOME FUNDAMENTAL CONCEPTS. THE SECOND SECTION DISCUSSES THE KINDS OF PROBLEMS THAT OCCUR IN COMPUTER NETWORKS. THE NEXT FOUR SECTIONS SURVEY METHODS OF PREVENTING, DETECTING, DIAGNOSING, AND RECOVERING FROM THEM. THE FINAL SECTION DESCRIBES SOME AREAS REQUIRING FURTHER STUDY.

A SURVEY OF TECHNIQUES FOR SYNCHRONIZATION AND RECOVERY IN DECENTRALIZED COMPUTER SYSTEMS
DOC. TYPE: JOURNAL
DOC. DATE: JUNE 1, 1981
AUTHORS: KOHLER, W. H.
SOURCE, PUB. BY, ETC: COMPUTING SURVEYS
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
TWO RELATED AND FUNDAMENTAL PROBLEMS IN DESIGNING DECENTRALIZED SYSTEMS WHICH SUPPORT AN OBJECT MODEL OF COMPUTATION ARE INTRODUCED, AND PROPOSED SOLUTION TECHNIQUES ARE SURVEYED. THE FIRST PROBLEM IS SYNCHRONIZING ACCESS TO SHARED OBJECTS WHILE ALLOWING A HIGH DEGREE OF CONCURRENCY. THE SECOND IS THE RECOVERY OF OBJECTS IN SPITE OF USER ERRORS, APPLICATION ERRORS, OR PARTIAL SYSTEM FAILURE. THE SYNCHRONIZATION PROBLEM IS A GENERALIZATION OF THE CONCURRENCY CONTROL PROBLEM WHICH ARISES IN DATABASE AND TRANSACTION-PROCESSING SYSTEMS. CONCURRENCY CONTROL METHODS WHICH USE LOCKING, TIMESTAMPS, CIRCULATING PERMIT, TUCKETS, CONFLICT ANALYSIS, AND RESERVATIONS ARE PRESENTED AND COMPARED. THE PROPOSED APPROACH TO SOLVING THE RECOVERY PROBLEM IS BASED ON A SOFTWARE STRUCTURING ABSTRACTION CALLED THE ATOMIC ACTION, A TYPE OF GENERALIZED TRANSACTION. REQUIREMENTS AND TECHNIQUES FOR IMPLEMENTING ATOMIC ACTIONS IN A DECENTRALIZED ENVIRONMENT ARE DISCUSSED.

A SYSTEMATIC APPROACH TO THE DESIGN OF FAULT TOLERANT DISTRIBUTED COMPUTER SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: COLEMAN, D. M. JORDAN, B. W., JR. SWAMY, S.
SOURCE, PUB. BY, ETC: IEEE
DOCUMENT NUMBER: IEEE# CH1864-8/83/0000/0128
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE DESIGN OF DISTRIBUTED COMPUTER SYSTEMS IS A COMPLEX TASK THAT NEEDS A DESIGN METHODOLOGY. WE PROPOSE A METHODOLOGY FOR TASK ALLOCATION BASED ON A CLUSTERING ALGORITHM CALLED THE UNIT MERGE ALGORITHM (UMA). TASK ALLOCATION IS CHARACTERIZED BY A MULTITUDE OF CONSTRAINTS AND OPTIMIZATIONS ONLY SOME OF WHICH MAY BE OF INTEREST TO A PARTICULAR DISTRIBUTED COMPUTER SYSTEM. A METHODLOLOGY FOR TASK ALLOCATON REQUIRES ALGORITHMS THAT WILL NOT SERIOUSLY DEGRADE IN THEIR PERFORMANCE UNDER ADD-ON CONSTRAINTS. WE CALL THIS FEATURE ROBUSTNESS. WE DESCRIBE ROBUST EXTENSIONS OF THE UMA THAT APPLY TO HETEROGENOUS DISTRIBUTED COMPUTER SYSTEMS. WE PROPOSE TWO TECHNIQUES FOR FAULT TOLERANT ALLOCATION AND SHOW HOW THE UMA AND ITS EXTENSIONS CAN BE USED TO IMPLEMENT THEM.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

A SYSTEMATIC APPROACH TO THE PERFORMANCE MODELING OF COMPUTER SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: 1979
AUTHORS: KEINZLE, M. SEVEIK, K.
SOURCE, PUB. BY, ETC: PERF OF COMPUTER SYSTS - ITASA, N. HOL PUB
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS PAPER PROPOSES A MODELING PROCEDURE THAT RELATES THE DIFFERENT ASPECTS OF MODELING: MODEL DESIGN, SYSTEM MEASUREMENT, PARAMETER ESTIMATION, MODEL ANALYSIS, MODEL VALIDATION, PERFORMANCE PREDICTION. THE MODELING PROCEDURE IS ORGANIZED IN SEVERAL STEPS THAT ISOLATE THE DIFFERENT ABSTRACTION STEPS TO PROVIDE A BETTER UNDERSTANDING OF THE MODELING PROCESS. THE PROPOSED PROCEDURE IS APPLIED TO A CASE STUDY.

A TAXONOMY OF TOOL FEATURES FOR THE ADA PROGRAMMING SUPPORT ENVIRONMENT (APSE)
DOC. TYPE: REPORT
AUTHORS: HOUGHTON, R.C.
FILED BY (NAME): KNOSP, A.

A TUTORIAL ON DISTRIBUTED PROCESSING IN AIRCRAFT/AVIONICS APPLICATIONS
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: ZEMPOLICH, B. A.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE, 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE PURPOSE OF THIS TUTORIAL IS TO PRESENT AN OVERVIEW OF THE STATE-OF-THE-ART IN REAL-TIME DISTRIBUTED PROCESSING AS APPLIED TO AIRCRAFT/AVIONICS. DEFINITIONS AND CONCEPTS ARE PRESENTED STARTING WITH THE TOTAL AIRCRAFT AS AREAL-TIME DISTRIBUTED COMPUTER-CONTROLLED SYSTEM. THE RELATIONSHIP OF AIR CRAFT MISSION AND AVIONIC SYSTEM ARCHITECTURES ID DISCUSSED. OVERALL SYSTEM ARCHITECTURAL CONSIDERATIONS ARE IDENTIFIED AND THEIR IMPACT UPON A REAL-TIME DISTRIBUTED COMPUTER-CONTROLLED SYSTEM IS DETAILED. A TOPDOWN HIERARCHICAL, ARCHITECTURAL STRUCTURE IS PRESENTED. THIS TOP-DOWN STRUCTURING IS DESCRIBED IN TERMS OF THE LOGICAL FUNCTIONAL DECOMPOSITION OF THE SYSTEM ASFOLLOWS: TOTAL AIRCRAFT/AVIONIC SYSTEM PARTITIONING OF AIRCRAFT/AVIONIC SUBSYSTEMS, INTERCONNECT BUS STRUCTURE (NETWORK), SYSTEM-WIDE PROCESSING ARCHITECTURE, SUBSYSTEMS DEFINITION, AND COMPUTER SYSTEMS.

THE AUTHOR IS DEPUTY TECHNOLOGY ADMINISTRATOR FOR COMMAND, CONTROL AND GUIDANCE, RESEARCH AND TECHNOLOGY GROUP. NAVAL AIR SYSTEMS COMMAND, WASH. D. C. 20361.

A UNIFIED METHOD FOR EVALUATING REAL-TIME COMPUTER CONTROLLERS: A CASE STUDY
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: SHINK, G. KRISHNA, C. LEE, Y.
SOURCE, PUB. BY, ETC: DEPT ELEC & COMPUTER ENGR - UNIV OF MICH
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
A REAL-TIME CONTROL SYSTEM CONSISTS OF A SYNERGISTIC PAIR, THAT IS, A CONTROLLED PROCESS AND A CONTROLLER COMPUTER. WE HAVE DEFINED NEW PERFORMANCE MEASURES FOR REAL-TIME CONTROLLER COMPUTERS ON THE BASIS OF THE NATURE OF THIS SYNERGISTIC PAIR.

IN THIS PAPER WE PRESENT A CASE STUDY OF A TYPICAL CRITICAL CONTROLLED PROCESS IN THE CONTEXT OF NEW PERFORMANCE MEASURES THAT EXPRESS THE PERFORMANCE OF BOTH CONTROLLED PROCESSES AND REAL-TIME CONTROLLERS (TAKEN AS A UNIT) ON THE BASIS OF A SINGLE VARIABLE: CONTROLLER RESPONSE TIME. CONTROLLER RESPONSE TIME IS A FUNCTION OF CURRENT SYSTEM STATE, SYSTEM FAILURE RATE, ELECTRICAL AND/OR MAGNETIC INTERFERENCE, ETC., AND IS THEREFORE A RANDOM VARIABLE. CONTROL OVERHEAD IS EXPRESSED AS A MONOCATASTROPHIC FAILURE, OR DYNAMIC FAILURE, IF THE RESPONSE TIME FOR A CONTROL TASK EXCEEDS THE CORRESPONDING SYSTEM HARD DEADLINE, IF ANY. A RIGOROUS PROBABILISTIC APPROACH IS USED TO ESTIMATE THE PERFORMANCE MEASURES.

A UNIFIED RELIABILITY MODEL FOR FAULT-TOLERANT COMPUTERS
DOC. TYPE: ARTICLE
DOC. DATE: NOVEMBER 1980
AUTHORS: NG, Y. AVIZIENIS, A.
SOURCE, PUB. BY, ETC: IEEE TRANS ON COMPUTERS, VOL C-29, NO.11
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE DIVERSIFIED NATURE OF FAULT-TOLERANT COMPUTERS LED TO THE DEVELOPMENT OF A MULTIPLICITY OF RELIABILITY MODELS WHICH ARE SEEMINGLY UNRELATED TO EACH OTHER. AS A RESULT, IT BECOMES DIFFICULT TO DEVELOP AUTOMATED TOOLS FOR RELIABILITY ANALYSIS WHICH ARE BOTH GENERAL AND EFFICIENT. THUS, THE POTENTIAL OF RELIABILITY MODELING AS A PRACTICAL AND USEFUL TOOL IN THE DESIGN PROCESS OF FAULT-TOLERANT COMPUTERS HAS NOT BEEN FULLY REALIZED. THIS PAPER SUMMARIZES THE RESULTS OF AN EXTENDED EFFORT TO DEVELOP A UNIFIED APPROACH TO RELIABILITY MODELING OF FAULT-TOLERANT COMPUTERS WHICH STRIKES A GOOD COMPROMISE BETWEEN

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

A UNIFIED RELIABILITY MODEL FOR FAULT-TOLERANT COMPUTERS * CONTINUED *

GENERALITY AND PRACTICALITY. THE UNIFIED MODEL DEVELOPED ENCOMPASSES REPAIRABLE AND NONREPAIRABLE SYSTEMS AND MODELS, TRANSIENT AS WELL AS PERMANENT FAULTS, AND THEIR RECOVERY. BASED ON THE UNIFIED MODEL, A POWERFUL AND EFFICIENT RELIABILITY ESTIMATION PROGRAM ARIES HAS BEEN DEVELOPED.

A VALIDATION PROTOTYPE OF SYSTEM RELIABILITY IN LIFE-CRITICAL APPLICATIONS

DOC. TYPE: PAPER
DOC. DATE: 1980
AUTHORS: TRIVEDI, K. GAULT, J. CLARY, J.
SOURCE, PUB. BY, ETC: PROC PATHWAYS TO SYS INTEGRITY SYMP - N.B.S.
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS PAPER ADDRESSES THE PROBLEM OF VALIDATING THE RELIABILITY OF COMPUTER SYSTEMS USED IN LIFE-CRITICAL APPLICATIONS. DUE TO EXTREMELY HIGH RELIABILITY REQUIREMENTS, TRADITIONAL VALIDATION METHODS BASED ON LIFETESTING ARE NO LONGER APPLICABLE. A VALIDATION APPROACH BASED ON A JUDICIOUS COMBINATION OF LOGICAL PROOFS, ANALYTICAL MODELS, AND EXPERIMENTAL TESTING IS ADVOCATED. THE ROLE OF MARKOV RELIABILITY MODELS IN THE VALIDATION PROCESS IS DISCUSSED AND A TAXONOMY OF VALIDATION TECHNIQUES IS PRESENTED.

A VALIDATION TECHNIQUE FOR TIGHTLY COUPLED PROTOCOLS

DOC. TYPE: ARTICLE
DOC. DATE: JULY 1982
AUTHORS: RUDIN, H. WEST, C.
SOURCE, PUB. BY, ETC: IEEE TRANS COMPUTERS, C-31, #7, JULY 82
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
RING SYSTEMS HAVE A NUMBER OF PROPERTIES THAT DISTINGUISH THEM FROM MORE GENERAL COMPUTER NETWORKS. CHANGES IN THE PROTOCAL MODELS AND VALIDATION TECHNIQUES ARE REPORTED ON. (AUTHORS ARE WITH IBM ZURICH LABORATORY, 8803 RUSCHLIKON. SWITZERLAND.)

A WORKLOAD DEPENDENT SOFTWARE RELIABILITY PREDICTION MODEL

DOC. TYPE: PAPER
DOC. DATE: JUNE 1982
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.
SOURCE, PUB. BY, ETC: 12TH ANN INTL SYMP ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IN THIS PAPER, THE EXPECTED NORMAL MANIFESTATIONS OF HARDWARE TRANSIENT FAULTS ARE CHARACTERIZED FOR TIME-SHARING SYSTEMS. A MODEL IS PRESENTED DERIVED FROM BASIC PRINCIPLES WHICH PREDICTS, FOR LONG OBSERVATION INTERVALS, THAT THE AVERAGE SYSTEM FAILURE RATE DUE TO TRANSIENTS AS A FUNCTION OF TIME OF DAY SHOULD BE A LINEAR FUNCTION OF THE AVERAGE FRACTION OF TIME SPENT IN KERNEL MODE. DEVIATIONS FROM THE EXPECTED NORMAL BEHAVIOR CAN BE USED TO DETERMINE THE CONTRIBUTION OF DESIGN ERRORS (IN PARTICULAR SOFTWARE ERRORS) TO SYSTEM UNRELIABILITY. THE MODEL HAS BEEN CALIBRATED FOR A REAL SYSTEM, FOR WHICH IT IS SHOWN HOW THE CONTRIBUTIONS OF HARDWARE TRANSIENTS AND SOFTWARE ERRORS TO SYSTEM UNRELIABILITY DEPEND ON SYSTEM WORKLOAD.

ADA -- AN ADVANCED INTRODUCTION

DOC. TYPE: BOOK
DOC. DATE: 1983
AUTHORS: GEHANI, NARAIN
SOURCE, PUB. BY, ETC: PRENTICE-HALL, INC. - ENGLEWOOD CLIFFS, N.J. 07632
DOCUMENT NUMBER: ISBN 0-13-003962-4
FILED BY (NAME): KNOSP, A.

ADA AND THE INTEL IAP-432 SUPPORT OF NETWORKS

DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1983
AUTHORS: MCKAY, C. W. BOWN, R. L. HART, D. A.
SOURCE, PUB. BY, ETC: AIAA COMPUTERS IN AEROSPACE IV CONFERENCE
DOCUMENT NUMBER: AIAA-83-2417-CP (NOT INCL IN CONF PROC'GS)
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER DESCRIBES FEATURES OF ADA, THE ADA PROGRAMMING SUPPORT ENVIRONMENT, DUALISM OF HARDWARE AND SOFTWARE, DUALISM OF ALGORITHMS AND DATA STRUCTURES, OBJECT ORIENTED DESIGN, THE INTEL IAPX432 CONCEPTS, AND THE BIU-OSI CONCEPTS OF A PROOF-OF-PRINCIPLE PORTION OF A DISTRIBUTED NETWORK PROJECT CURRENTLY UNDERWAY AMONG NASA/JSC, UH-CL, AND PRIVATE INDUSTRY PERSONNEL.

THE AUTHORS ARE WITH THE HIGH TECHNOLOGIES LABORATORY, UNIVERSITY OF HOUSTON - CLEAR LAKE, HOUSTON, TX.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

ADVANCED AVIONICS SYSTEM FOR AN AEROSPACE PLANE
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: JORGENSEN, D.E. HERNANDEZ, C.M.
SOURCE, PUB. BY, ETC: PROC IEEE/AIAA 5TH DIG AV SYS CONF, 10/31-11/03/83
FILED BY (NAME): WERNER, R.
ABSTRACT:
THE AVIONICS SYSTEM FOR AN ADVANCED, MANNED AEROSPACE PLANE (ASP) WILL REQUIRE THE INTEGRATION OF SOPHISTICATED EQUIPMENT TO MEET THE MISSION REQMNTS OF THE 1990'S. ALTHOUGH THESE MISSION REQMNTS HAVE NOT BEEN EXPLICITLY DEFINED, LIKE THE AIRPLANE WITH ITS BROAD SPECTRUM OF CAPABILITY, THE ASP COULD BE REQUIRED TO PERFORM TRANSPORTATION TASKS FOR PASSENGERS, SOPHISTICATED EQUIPMENT, AND/OR SPACE BASE SUPPLIES; & PERFORM SCIENTIFIC & RESCUE MISSIONS. THE REQUIREMENTS FOR AN ASP CAN BE HYPOTHESIZED BASED ON REQMNTS THAT ARE UNIVERSAL TO POST-SPACE TRANSPORTATION SYSTEM VEHICLES, & REQUIREMENTS THAT ARE PRESENTLY BEING RESOLVED. AN AVIONICS SYSTEM TO SUPPORT AN ASP WILL INCLUDE THE CAPABILITY TO FLY THE MISSION IN A MANNED AUTOMATIC AND/OR MANUAL MODE, CHANGE MISSION OBJECTIVES AFTER LAUNCH, PERFORM MISSION WITHOUT GROUND AID, RESPONDING TO A 24-HOUR LAUNCH DEMAND AS WELL AS MEETING A 48-HOUR TURNAROUND TIME, & SUPPORT THE PAYLOAD WITH HEALTH STATUS THRUPUT/PAYLOAD DEPLOYMENT/SENSOR OPERATIONS. THIS PAPER IDENTIFIES THE FUNCTIONAL SUBSYSTEM REQUIREMENTS, REVIEWS THE CRITICAL ISSUES, PROJECTS THE TECHNOLOGY INTO THE 1990'S, & DELINEATES THE PHYSICAL CHARACTERISTICS OF THE SYSTEM.

ADVANCED RELIABILITY MODELING OF FAULT-TOLERANT COMPUTER-BASED SYSTEMS
DOC. TYPE: TECH MEMO
DOC. DATE: MAY 1982
AUTHORS: BAVUSO, S. J.
SOURCE, PUB. BY, ETC: NASA LANGLEY RESEARCH CENTER
DOCUMENT NUMBER: NASA TM-84501
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IT IS THE INTENT OF THIS PRESENTATION TO ADDRESS THE NUANCES OF MODELING THE RELIABILITY OF SYSTEMS WITH LARGE STATE SIZES, IN THE "MARKOV" SENSE, WHICH RESULT FROM SYSTEMS THAT ARE BASED ON REPLICATED REDUNDANT HARDWARE AND TO DISCUSS THE MODELING OF NUMEROUS FACTORS WHICH CAN REDUCE RELIABILITY WITHOUT CONCOMITANT DEPLETION OF SPARE HARDWARE. THE DIMINISHING FACTORS ARE CAPTURED BY THE POPULAR "COVERAGE" TERMINOLOGY. ADVANCED COVERAGE (FAULT-HANDLING) MODELS ARE DESCRIBED WITH SUPPORTING RATIONALE. METHODS OF ACQUIRING AND MEASURING PARAMETERS FOR THESE MODELS ARE DELINEATED, AND SOME RECENTLY MEASURED LATENT-FAULT DATA ARE PRESENTED.

AIPS SYSTEM REQUIREMENTS
DOC. TYPE: CSDL REPORT
DOC. DATE: AUGUST 30, 1983
AUTHORS: FELLEMAN, P. G.
SOURCE, PUB. BY, ETC: CSDL, AIPS PROGRAM
DOCUMENT NUMBER: AIPS-83-50
FILED BY (NAME): FELLEMAN, P. G.
ABSTRACT:
THE EXPECTED RANGES OF COMMON REQUIREMENTS ACROSS APPLICATIONS OF THE NASA ADVANCED INFORMATION PROCESSING SYSTEM (AIPS) ARE CHARACTERIZED. TOP-LEVEL SYSTEM REQUIREMENTS ARE IDENTIFIED, AND ARE INTENDED TO BE APPLICATION AND IMPLEMENTATION INDEPENDENT. THEY INCLUDE SUBJECTIVE DESIGN OBJECTIVES, QUANTIFIABLE PERFORMANCE RANGES, AND SELECTED FUNCTIONAL CHARACTERISTICS.

THE REPORT CONTAINS A DESCRIPTION OF THE APPROACH USED TO GENERATE THE REQUIREMENTS, THE RESULTS OF THE AIPS APPLICATION REQUIREMENTS SURVEY, A DISCUSSION OF ANTICIPATED GROWTH AND ITS ACCOMMODATION, A SUMMARY OF THE SPECTRUM OF COMMON APPLICATION REQUIREMENTS DERIVED FROM THE SURVEY, AND THE TOP-LEVEL AIPS SYSTEM REQUIREMENTS. THE EMPHASIS IN THE REQUIREMENTS DEFINITION EFFORT WAS ON FUNCTIONAL DESCRIPTION, NOT ON DESIGN APPROACHES. THE GOAL WAS TO DEVELOP AIPS TOP-LEVEL SYSTEM REQUIREMENTS THAT REFLECT THE ADVANCED PROCESSING AND CONTROL REQUIREMENTS FOR MAJOR AEROSPACE APPLICATIONS IN THE EARLY 1990'S.

AIPS TECHNOLOGY SURVEY
DOC. TYPE: REPORT
DOC. DATE: AUGUST 26, 1983
SOURCE, PUB. BY, ETC: CSDL TIC
FILED BY (NAME): WERNER, R.E.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

AN APPROACH TO THE ORGANIZATION OF KNOWLEDGE AND ITS USE IN NATURAL LANGUAGE RECALL TASKS
DOC. TYPE: ARTICLE
DOC. DATE: 1983
AUTHORS: MCCALLA, G. I.
SOURCE, PUB. BY, ETC: COMPUTERS & MATH, WITH APPLICS., (GB), V 9, #1
DOCUMENT NUMBER: ISSN 0097-4943
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE VIEWPOINT ESPOUSED IN THIS PAPER IS THAT NATURAL LANGUAGE UNDERSTANDING AND PRODUCTION IS THE ACTION OF A NUMBER OF HIGHLY INTEGRATED DOMAIN-SPECIFIC SPECIALISTS. DESCRIBED FIRST IS AN OBJECT ORIENTED REPRESENTATION SCHEME WITH ALLOWS THESE SPECIALISTS TO BE BUILT. DISCUSSED NEXT IS THE ORGANIZATION OF THESE SPECIALISTS INTO A FOUR-LEVEL GOAL HIERARCHY THAT ENABLES THE MODELLING OF NATURAL LANGUAGE CONVERSATION. IT IS SHOWN HOW THE REPRESENTATION AND NATURAL LANGUAGE STRUCTURES CAN BE USED TO FACILITATE THE RECALL OF EARLIER NATURAL LANGUAGE CONVERSATIONS. SIX SPECIFIC KINDS OF RECALL TASKS ARE OUTLINED IN TERMS OF THESE STRUCTURES, AND THEIR OCCURRENCE IN SEVERAL LEGAL DIALOGUES IS EXAMINED. FINALLY, THENEED FOR INTELLIGENT GARBAGE COLLECTION OF OLD EPISODIC INFORMATION IS POINTED OUT.

ANALYSIS OF M/G/2 - STANDBY REDUNDANT SYSTEM
DOC. TYPE: PAPER
AUTHORS: BACCELLI, F. TRIVEDI, K.
SOURCE, PUB. BY, ETC: INRIA, FRANCE; DUKE UNIVERSITY, U.S.A.
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
DEGRADABLE SYSTS CAN NO LONGER BE ADEQUATELY CHARACTERIZED BY A SEPARATION OF RELIABILITY/AVAILABILITY MEASURES FROM PERFORMANCE (THRUPUT/RESP TIME) MEASURES. IT IS NECESSARY TO DERIVE COMPOSITE MEASURES OF SYS EFFECTIVENESS. SUCH COMPOSITE MEASURES HAVE BEEN VARIOUSLY CALLED PERFORMABILITY, COMPUTATION-BASED REL/AVAIL, OR PERF REL.

MUCH OF THE PRIOR WORK ON THE COMBINED EVAL OF PERF AND REL IS BASED ON THE OBSERVATION THAT TIMES TO FAILURE AND REPAIR ARE SEVERAL ORDERS OF MAG LARGER THAN TIME TO COMPLETE A JOB. THEREFORE, SYS EFFECTIVENESS CAN BE STUDIED BY DECOMPOSING ITS BEHAVIOR INTO SUBMODELS AND LATER COMBINING THE SUBMODELS. WHILE THIS ASSUMP IS GENERALLY VALID AND THE RESULTING ANSWERS ARE REASONABLY ACCURATE, IT IS HOWEVER DESIRABLE TO DEV EXACT (RATHER THAN APPROX) MODELS FOR SYS EFFECTIVENESS.

WE CONSIDER A 2-UNIT STANDBY REDUNDANT SYS WITH FAILURE AND REPAIR. TIMES TO FAILURE OF COMPS AND REPAIR ARE EXPONENTIALLY DISTRIBUTED. THE JOB ARRIVAL PROCESS IS POISSON WHILE THE JOB SERVICE TIME DISTRIBUTION IS ASSUMED TO BE ABSOLUTELY CONTINUOUS. A FAIL DISRUPTS THE SERVICE OBTAINED BY THE JOB UNDERGOING SERV, SO THAT JOB HAS TO BE RESTARTED WHENEVER THE SERVICE IS AVAIL.

ANALYTIC QUEUEING THEORY FOR PROGRAMS WITH INTERNAL CONCURRENCY
DOC. TYPE: ARTICLE
DOC. DATE: 1983
AUTHORS: HEIDELBERGER, P. TRIVEDI, K.
SOURCE, PUB. BY, ETC: IEEE TRANS ON COMPUTERS, VOL C-32, NO.1
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
ANALYTIC QUEUEING MODELS OF PROGRAMS WITH INTERNAL CONCURRENCY ARE CONSIDERED. THE PROGRAM BEHAVIOR MODEL ALLOWS A PROCESS TO SPAWN TWO OR MORE CONCURRENT TASKS AT SOME POINT DURING ITS EXECUTION. EXCEPT FOR QUEUEING EFFECTS, THE TASKS EXECUTE INDEPENDENTLY OF ONE ANOTHER, AND AT THE END OF THEIR EXECUTION, EITHER WAIT FOR ALL OF THEIR SIBLINGS TO FINISH EXECUTION OR MERGE WITH THE PARENT IF ALL HAVE FINISHED EXECUTION. TWO APPROXIMATE SOLUTION METHODS FOR THE PERFORMANCE PREDICTION OF SUCH SYSTEMS ARE DEVELOPED, AND RESULTS OF THE APPROXIMATIONS ARE COMPARED TO THOSE OF SIMULATIONS. THE APPROXIMATIONS ARE BOTH COMPUTATIONALLY EFFICIENT AND HIGHLY ACCURATE. THE GAIN IN PERFORMANCE DUE TO MULTITASKING AND MULTIPROCESSING IS STUDIED WITH A SERIES OF EXAMPLES.

ARIES 81 - A RELIABILITY AND LIFE CYCLE EVALUATION TOOL FOR FAULT-TOLERANT SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: JUNE 1982
AUTHORS: MAKAM, S. AVIZIENIS, A.
SOURCE, PUB. BY, ETC: 12TH ANN INT SYMP ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS PAPER DESCRIBES A PROGRAM CALLED ARIES 81 (AUTOMATED RELIABILITY INTERACTIVE ESTIMATION SYSTEM) WHICH IS DESIGNED TO PROVIDE STUDENTS, RESEARCHERS, AND DESIGNERS WITH AN ENVIRONMENT IN WHICH THEY MAY DEFINE AND ANALYZE THE RELIABILITY MODELS FOR FAULT-TOLERANT COMPUTERS. ARIES 81 IS AN EXTENDED AND IMPROVED VERSION OF A RELIABILITY ESTIMATION PROGRAM CALLED ARIES 76. THE USER OF ARIES 81 IS PRESENTED WITH A MATHEMATICAL FRAMEWORK OF ANALYSIS -- STOCHASTIC HOMOGENEOUS MARKOV PROCESS, IN WHICH TO MODEL THE SYSTEM HE IS DESIGNING. ARIES 81 HAS THE CAPABILITY TO CONSTRUCT, PROCESS, AND EVALUATE THE PREDEFINED SPECIFIC MODELS FOR CLOSED, REPAIRABLE, AND PERIODICALLY RENEWED CLOSED FAULT-TOLERANT SYSTEMS. IT CAN ALSO PROCESS AND EVALUATE ANY NEWLY DEFINED AND USER

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

ARIES 81 - A RELIABILITY AND LIFE CYCLE EVALUATION TOOL FOR FAULT-TOLERANT SYSTEMS * CONTINUED *

CONSTRUCTED MODELS FOR MORE COMPLEX SYSTEMS AND CAN MODEL BOTH PERMANENT AND TRANSIENT FAULT RECOVERY. BOTH RELIABILITY AND LIFE-CYCLE ANALYSIS CAN BE PERFORMED WITH THE AID OF ARIES 81.

AUTOMATED RELIABILITY AND FAILURE EFFECTS METHODS FOR DIGITAL FLIGHT CONTROL AND AVIONICS SYSTEMS - VOLUME 1: EVALUATION

DOC. TYPE: REPORT
DOC. DATE: MARCH 1981
AUTHORS: NESS, W. MCCRARY. W. BRIDGEMAN, M.
SOURCE, PUB. BY, ETC: LOCKHEED-GEORGIA CO - BATTELLE COLUMBUS LAB
DOCUMENT NUMBER: NASA CR - 166148
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE PRIMARY OBJECTIVE OF THIS STUDY WAS TO SELECT THE BEST EXISTING METHODS FOR PREDICTING THE RELIABILITY AND FAILURE EFFECTS OF DIGITAL FLIGHT CONTROL AND AVIONICS SYSTEMS (DFCAS). TWELVE RELIABILITY PREDICTION METHODS AND ONE FAILURE EFFECTS METHOD WERE IDENTIFIED FOR EVALUATION. OF THE TWELVE RELIABILITY PREDICTION MODELS, EIGHT WERE ELIMINATED BECAUSE OF MAJOR DEFICIENCIES WITH RESPECT TO CONFIGURATION ADAPTABILITY, ABILITY TO ANALYZE TRANSIENT FAULTS, USER IMPACTS, AND SCHEDULE CONSIDERATIONS. THE REMAINING FOUR METHODS, ARIES, CARE II, CARSA, AND CAST, WERE INVESTIGATED IN DETAIL.

BABELISM, BABBAGE'S BOOSTER, AND BERNOULLI'S NUMBERS

DOC. TYPE: REPORT
AUTHORS: DACOSTA, ROBERT
FILED BY (NAME): KNOSP, A.

BEST/1 - DESIGN OF A TOOL FOR COMPUTER SYSTEM CAPACITY PLANNING

DOC. TYPE: PAPER
DOC. DATE: 1979
AUTHORS: BUZEN, J.
SOURCE, PUB. BY, ETC: NATIONAL COMPUTER CONFERENCE
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS PAPER PRESENTS THE DESIGN PHILOSOPHY USED IN THE BEST/1 COMPUTER PERFORMANCE EVALUATOR. ESSENTIALLY, BEST/1 IS A MODELING TOOL WHICH HAS BEEN DEVELOPED TO ADDRESS THE "WHAT IF" QUESTIONS THAT ARISE WHEN EVALUATING COMPUTER SYSTEM PERFORMANCE. THESE QUESTIONS ARE OF IMPORTANCE IN SUCH AREAS AS CAPACITY PLANNING, PERFORMANCE TUNING, HARDWARE VENDOR SELECTION AND NEW SYSTEM DESIGN.

BOEING 757 AND 767 FLIGHT MANAGEMENT SYSTEM

DOC. TYPE: PAPER
DOC. DATE: NOVEMBER 20, 1980
AUTHORS: SPRADLIN, R.
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THE MAIN FEATURES WHICH CHARACTERIZE THE FMS ARE THE FEDERATED SYSTEM ARCHITECTURE AND THE SYSTEM INTERFACE WITH THE FLIGHT CREW. THE BASIC FMS ARCHITECTURE WAS DEVELOPED OVER SEVERAL YEARS, IS RESPONSIVE TO A BROAD RANGE OF AIRLINE DESIRES AND PROVIDES THE OPERATIONAL FLEXIBILITY ESSENTIAL TO OPERATIONS ANYWHERE IN THE WORLD. THE SYSTEM INTERFACE WITH THE PILOT IS THE PRODUCT OF EXTENSIVE DEVELOPMENT. THE FMS IS DESIGNED TO INCREASE THE EFFECTIVENESS OF THE PILOT IN HIS ROLE AS FLIGHT MANAGER BY PROVIDING READY ACCESS TO RELEVANT FLIGHT INFORMATION AND PROVIDES SUITABLE MEANS TO ACCOMPLISH AIRCRAFT CONTROL WITHIN REASONABLE WORKLOAD BOUNDS.

CARE III FINAL REPORT PHASE I VOLUME I

DOC. TYPE: REPORT
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.
SOURCE, PUB. BY, ETC: RAYTHEON CO.
DOCUMENT NUMBER: NASA CR -159122
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS RPT DESCRIBES THE WORK DONE DURING THE 1ST PHASE OF A 2-PHASE EFFORT TO DEV A COMPUTER PROG TO AID IN ASSESSING THE RELIABILITY OF FAULT-TOLERANT AVIONICS SYSTS. THE OVERALL EFFORT CONSISTS OF 5 MAJOR TASKS: 1) ESTABLISH THE BASIC REQMTS THAT MUST BE SATISFIED IF THE PROG IS TO ACHIEVE ITS OVERALL OBJ. 2) DEFINE A GEN PROG STRUCTURE CONSISTENT WITH THESE REQMTS. 3) DEV & PROG A MATHEMATICAL MODEL RELATING THE RELIABILITY OF A FAULT-TOLERANT SYS TO THE FAIL RATES & COVERAGE FACTORS CHARACTERIZING ITS VARIOUS ELEMENTS. 4) DEV & PROG A MATHEMATICAL MODEL FOR EVAL THE COVERAGE ASSOC WITH ANY GIVEN FAULT AS A FUNC OF THE TYPE & LOCATION OF THE FAULT, THE APPLICABLE FAULT DETECTION & ISOLATION MECHANISM, & THE NO. & STAT OF PRIOR FAULTS. 5) DEV & PROG A PROC WHEREBY A USER OF THESE MODELS CAN ACCURATELY & CONVENIENTLY SPECIFY THE CONFIG OF THE SYS TO BE EVAL & THE CONSTRAINTS INFLUENCING ITS ABILITY TO RECOVER FROM FAULTS.

THE 1ST 3 OF THESE TASKS WERE COMPLETED DURING PHASE 1; THE RESULTING REQMTS PROG

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

CARE III FINAL REPORT PHASE I VOLUME I * CONTINUED *

STRUCTURE, & RELIABILITY MODEL ARE DISCUSSED IN DETAIL IN VOL1 OF THIS RPT, ALONG WITH THE TRADEOFFS & SAMPLE RELIABILITY ASSESSMNTS MADE IN ARRIVING AT THE APPROACH FINALLY TAKEN.

CARE III FINAL REPORT PHASE I VOLUME II

DOC. TYPE: REPORT
DOC. DATE: NOVEMBER 1979
AUTHORS: STIFFLER, J. BRYANT, L. GUCCIONE, L.
SOURCE, PUB. BY, ETC: RAYTHEON CO.
DOCUMENT NUMBER: NASA CR-159123
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS RPT DESCRIBES THE WORK DONE DURING THE 1ST PHASE OF A 2-PHASE EFFORT TO DEV A COMP PROG TO AID IN ASSESSING THE RELIABILITY OF FAULT-TOLERANT AVIONICS SYSTS. THE OVERALL EFFORT CONSISTS OF 5 MAJOR TASKS: 1)ESTABLISH THE BASIC REQMNTS THAT MUST BE SATISFIED IF THE PROG IS TO ACHIEVE ITS OVERALL OBJ. 2)DEFINE A GEN PROG STRUCTURE CONSISTENT WITH THESE REQMNTS. 3)DEV & PROG A MATHEMATICAL MODEL RELATING THE RELIABILITY OF A FAULT-TOLERANT SYS TO THE FAILURE RATES & COVERAGE CHARACTERIZING ITS VARIOUS ELEMENTS. 4)DEV & PROG A MATHEMATICAL MODEL FOR EVAL THE COVERAGE ASSOC WITH ANY GIVEN FAULT AS A FUNC OF THE TYPE & LOCATION OF THE FAULT, THE APPLICABLE FAULT DETECTION & ISOLATION MECHANISM & THE NUMBER & STATUS OF PRIOR FAULTS. 5)DEV & PROG A PROC WHEREBY A USER OF THESE MODELS CAN ACCURATELY & CONVENIENTLY SPECIFY THE CONFIG OF THE SYS TO BE EVAL & THE CONSTRAINTS INFLUENCING ITS ABILITY TO RECOVER FROM FAULTS.

THE 1ST 3 OF THESE TASKS WERE COMPLETED DURING PHASE 1; THE RESULTING REQMNTS, PROG STRUCTURE, & RELIABILITY MODEL ARE DISCUSSED IN DETAIL IN VOL1 OF THIS RPT, ALONG WITH THE TRADEOFFS & SAMPLE RELIABILITY ASSESSMENTS MADE IN ARRIVING AT THE APPROACH FINALLY TAKEN.

CARE III PHASE II REPORT - MATHEMATICAL DESCRIPTION

DOC. TYPE: REPORT
DOC. DATE: NOVEMBER 1982
AUTHORS: STIFFLER, J. BRYANT, L.
SOURCE, PUB. BY, ETC: RAYTHEON CO.
DOCUMENT NUMBER: NASA CR-3566
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
CARE III (COMPUTER-AIDED RELIABILITY EST, VERS3) IS A COMP PROGDESIGNED TO HELP EST THE RELIABILITY OF COMPLEX, REDUNDANT SYSTS, ALTHOUGH THE PROG CAN MODEL A WIDE VARIETY OF REDUNDANT STRUCTURES, IT WAS DEV SPECIFICALLY FOR FAULT-TOLERANT AVIONICS SYSTS; SYSTS DISTINGUISHED BY THE NEED FOR XTREMELY RELIABLE PERF SINCE A SYS FAIL COULD WELL RESULT IN THE LOSS OF HUMAN LIFE.

THE 1ST CARE PROG, DEV AT THE JET PROPUL LAB IN 1971, PROVIDED AN AID FOR EST THE RELIABILITY OF SYSTS CONSISTING OF A COMBO OF ANY OF SEVERAL STAND CONFIGS (E.G., STANDBY-REPLACEMNT & TRIPLE-MODULAR REDUNDANT CONFIGS, ETC). CARE II WAS SUBSEQUENTLY DEV BY RAYTHEON, UNDER CONTRACT TO THE NASA LANG RES CTR, IN 1974. IT SUBSTANTIALLY GENERALIZED THE CLASS OF REDUNDANT CONFIGS THAT COULD BE ACCOMMODATED, & INCL A COVERAGE MODEL TO DETERMINE THE VARIOUS COVERAGE PROBABILITIES AS A FUNC OF THE APPLICABLE FAULT RECOVERY MECHANISMS (DETECTION DELAY, DIAG SCHED INTERVAL, ISOLATION & RECOVERY DELAY, ETC).

CARE III FURTHER GENERALIZES THE CLASS OF SYS STRUCTURES THAT CAN BE MODELED & GREATLY EXPANDS THE COVERAGE MODEL TO TAKE INTO ACCT SUCH EFFECTS AS INTERMITTENT, TRANSIENT, & LATENT FAULTS, ERROR PROPAGATION, ETC.

CARE III PHASE II REPORT USER'S MANUAL

DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1982
AUTHORS: BRYANT, L. STIFFLER, J.
SOURCE, PUB. BY, ETC: RAYTHEON CO.
DOCUMENT NUMBER: NASA CR-165864
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
CARE III (COMP-AIDED RELIABILITY ESTIMATION, VERS3) IS A COMP PROG DESIGNED TO HELP ESTIMATE THE RELIABILITY OF COMPLEX, REDUNDANT SYSTS. ALTHOUGH THE PROG CAN MODEL A WIDE VARIETY OF REDUNDANT STRUCTURES, IT WAS DEV SPECIFICALLY FOR FAULT-TOLERANT AVIONICS SYSTS--SYSTS DISTINGUISHED BY THE NEED FOR XTREMELY RELIABLE PERF SINCE A SYS FAIL COULD WELL RESULT IN THE LOSS OF HUMAN LIFE.

THE 1ST CARE PROG, DEV AT THE JET PROPUL LAB IN 1971, PROVIDED AN AID FOR ESTIMATING THE RELIABILITY OF SYSTS CONSISTING OF A COMBO OF ANY OF SEVERAL STAND CONFIGS (E.G, STANDBY-REPL & TRIPLE-MODULAR REDUNDANT CONFIGS, ETC). CARE II WAS SUBSEQUENTLY DEV BY RAYTHEON, UNDER CONTRACT TO NASA LANGLEY RES CTR, IN 1974. IT SUBSTANTIALLY GENERALIZED THE CLASS OF REDUNDANT CONFIGS THAT COULD BE ACCOMMODATED, & INCL A COVERAGE MODEL TO DETERMINE THE VARIOUS COVERAGE PROBABILITIES AS A FUNC OF THE APPLICABLE FAULT RECOVERY MECHANISMS (DETECTION, ISOLATION & RECOVER DELAY, & DIAG SCHED INTERVAL, ETC).

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

CARE III PHASE II REPORT USER'S MANUAL * CONTINUED *

CARE III FURTHER GENERALIZES THE CLASS OF SYS STRUCTURES THAT CAN BE MODELED & GREATLY EXPANDS THE COVERAGE MODEL TO TAKE INTO ACCT SUCH EFFECTS AS INTERMITTENT, TRANSIENT & LATENT FAULTS, ERROR PROPAGATION, ETC.

CARE III PHASE III REPORT - TEST AND EVALUATION

DOC. TYPE: REPORT
DOC. DATE: NOVEMBER 1982
AUTHORS: STIFFLER, J. NEUMANN, J. BRYANT, L.
SOURCE, PUB. BY, ETC: RAYTHEON CO.
DOCUMENT NUMBER: NASA CR - 3631
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:

CARE III (COMPUTER-AIDED RELIABILITY ESTIMATION, VERS3) IS A COMPUTER PROG DESIGNED TO HELP EST THE RELIABILITY OF COMPLEX, REDUNDANT SYSTS; ALTHOUGH THE PROG CAN MODEL A WIDE VARIETY OF REDUNDANT STRUCTURES, IT WAS DEV SPECIFICALLY FOR FAULT-TOL AVIONICS SYSTS; SYSTS DISTINGUISHED BY THE NEED FOR XTREMELY RELIABLE PERF SINCE A SYS FAIL COULD WELL RESULT IN THE LOSS OF HUMAN LIFE.

THE 1ST CARE PROG, DEV AT THE JET PROPUL LAB IN 1971, PROVIDED AN AID FOR EST THE RELIABILITY OF SYSTS CONSISTING OF A COMBO OF ANY OF SEVERAL STAND CONFIGS (E.G., STANDBY-REPLACEMENT & TRIPLE-MODULAR REDUNDANT CONFIGS, ETC). CARE II WAS SUBSEQUENTLY DEV BY RAYTHEON, UNDER CONTRACT TO NASA LANGLEY CTR, IN 1974. IT SUBSTANTIALLY GENERALIZED THE CLASS OF REDUNDANT CONFIGS THAT COULD BE ACCOMMODATED, & INCL A COVERAGE MODEL TO DETERMINE THE VARIOUS COVERAGE PROBABILITIES AS A FUNC OF THE APPLICABLE FAULT RECOVERY MECHANISMS (DETECTION DELAY, DIAG SCHED INTERVAL, ISOLATION & RECOVERY DELAY, ETC).

CARE III FURTHER GENERALIZES THE CLASS OF SYS STRUCTURES THAT CAN BE MODELED & GREATLY EXPANDS THE COVERAGE MODEL TO TAKE INTO ACCT SUCH EFFECTS AS INTERMITTENT, TRANSIENT & LATENT FAULTS & ERROR PROPOGATION, ETC

CLOSED FORM SOLUTION OF PERFORMABILITY

DOC. TYPE: ARTICLE
DOC. DATE: JULY 1982
AUTHORS: MEYER, J. F.
SOURCE, PUB. BY, ETC: IEEE TRANS. ON COMPUTERS, VOL C-31, NO. 7
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:

IF COMPUTING SYSTEM PERFORMANCE IS DEGRADABLE, THEN AS RECOGNIZED IN A NUMBER OF RECENT STUDIES, SYSTEM EVALUATION MUST DEAL SIMULTANEOUSLY WITH ASPECTS OF BOTH PERFORMANCE AND RELIABILITY. ONE APPROACH IS THE EVALUATION OF A SYSTEM 'S "PERFORMABILITY," WHICH RELATIVE TO A SPECIFIED PERFORMANCE VARIABLE Y, GENERALLY REQUIRES SOLUTION OF THE PROBABILITY DISTRIBUTION FUNCTION OF Y. IN THIS PAPER WE EXAMINE THE FEASIBILITY OF CLOSED-FORM SOLUTIONS OF PERFORMABILITY WHEN Y IS CONTINUOUS. IN PARTICULAR, WE CONSIDER THE MODELING OF A DEGRADABLE BUFFER/MULTIPROCESSOR SYSTEM WHOSE PERFORMANCE Y IS THE (NORMALIZED) AVERAGE THROUGHPUT RATE REALIZED DURING A BOUNDED INTERVAL OF TIME. EMPLOYING AN APPROXIMATE DECOMPOSITION OF THE MODEL, WE SHOW THAT A CLOSED-FORM SOLUTION CAN INDEED BE OBTAINED.

COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS

DOC. TYPE: REPORT
DOC. DATE: APRIL 1981
AUTHORS: HITT, E. F. BRIDGEMAN, M. ROBINSON, A.
SOURCE, PUB. BY, ETC: (BATELLE COLUMBUS LABS.) NASA
DOCUMENT NUMBER: NASA CR-159358
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:

THE OBJECTIVE OF THIS STUDY WAS TO EVALUATE "PERFORMABILITY", A TECHNIQUE DEVELOPED BY THE UNIVERSITY OF MICHIGAN UNDER NASA GRANT NSG 1306, FOR ITS ACCURACY, PRACTICAL USEFULNESS, AND COST OF USE. PERFORMABILITY ANALYSIS DETERMINES THE PROBABILITIES OF OCCURRENCE FOR A SET OF MISSION OUTCOMES. IT WAS DESIGNED FOR APPLICATION TO FAULT-TOLERANT COMPUTING SYSTEMS USED IN MULTIPHASE MISSIONS. PERFORMABILITY WAS FOUND TO REQUIRE SIGNIFICANTLY MORE TIME TO LEARN AND UNDERSTAND THAN THE FAULT-TREE METHOD.

COMPARISON OF PROGRAMMING LANGUAGES: ADA, PRAXIS, PASCAL, C

DOC. TYPE: REPORT
DOC. DATE: 1981
AUTHORS: EVANS, A. JR.
SOURCE, PUB. BY, ETC: BOLT, BERANEK AND NEWMAN, INC. - CAMBRIDGE, MA
DOCUMENT NUMBER: UCRL-15346
FILED BY (NAME): KNOSP, A.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

COMPLETENESS CRITERIA FOR TESTING ELEMENTARY PROGRAM FUNCTIONS
DOC. TYPE: ARTICLE
DOC. DATE: SEPTEMBER 1981
AUTHORS: HOWDEN, W.
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
PROGRAM TESTING METRICS ARE BASED ON CRITERIA FOR MEASURING THE COMPLETENESS OF A SET OF PROGRAM TESTS. BRANCH TESTING MEASURES THE PERCENTAGE OF PROGRAM BRANCHES THAT ARE TRAVERSED DURING A SET OF TESTS. MUTATION TESTING MEASURES THE ABILITY OF A SET OF TESTS TO DISTINGUISH A PROGRAM FROM SIMILAR PROGRAMS. A CRITERION FOR TEST COMPLETENESS IS INTRODUCED IN THIS PAPER WHICH MEASURES THE ABILITY OF A SET OF TESTS TO DISTINGUISH BETWEEN FUNCTIONS WHICH ARE IMPLEMENTED BY PARTS OF PROGRAMS. THE CRITERION IS APPLIED TO FUNCTIONS WHICH ARE IMPLEMENTED BY DIFFERENT KINDS OF PROGRAMMING LANGUAGE STATEMENTS. IT IS MORE EFFECTIVE THAN BRANCH TESTING AND INCORPORATES SOME OF THE ADVANTAGES OF MUTATION TESTING. IT'S EFFECTIVENESS CAN BE DISCUSSED FORMALLY AND IT CAN BE DESCRIBED AS PART OF AN INTEGRATED APPROACH TO TESTING. A TOOL CAN BE USED TO IMPLEMENT THE METHOD.

COMPUTER FAMILY ARCH SEL COMM-FINAL RPT-V5-PROC FOR AND RESULTS OF THEEVAL UATION OF SOFTWARE BASES OF THE CANDIDATE ARCH FOR THE MIL COMPUTER FAMILY
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: LIEBLEIN, E. WAGNER, J. CLEARWATERS, A.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4530
FILED BY (NAME): MOTYKA, P. R.

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT - VOL VI - LIFE CYCLE COST MODELS
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: COLEMAN, A. CORNYN, J. SVIRSKY, W.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4535
FILED BY (NAME): MOTYKA, P. R.

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL III - EVALUATION OF COMPUTER ARCHITECTURE VIA TEST PROGRAMS
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. SHAMAN, P.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4528
FILED BY (NAME): MOTYKA, P. R.

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOL 1 - INTRODUCTION
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. COLEMAN, A. SMITH, R.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4526
FILED BY (NAME): MOTYKA, P. R.

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME II - SELECTION OF CANDIDATE ARCHITECTURES AND INITIAL SCREENING
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: BURR, W. FULLER, S. STONE, H.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4527
FILED BY (NAME): MOTYKA, P. R.

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME VIII - CFA FINAL SELECTION
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: CLEARWATERS, A. GORDON, R. SIEWIOREK, DANIEL P.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4531
FILED BY (NAME): MOTYKA, P. R.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IV-ARCHITECTURE RESEARCH FACILITY, ISP DESCRIPTION, SIMULATIONS, DATA COLLECTIONS
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: GORDON, R. HOWBRIGG, R. BUCKERMAN, S.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4529
FILED BY (NAME): MOTYKA, P. R.

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE-FINAL REPORT, VOL IX - A CONSIDERATION OF ISSUES IN THE SELECTION OF A COMPUTER FAMILY ARCHITECTURE
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1977
AUTHORS: CONRAD, T. ESTELL, R. HAYNES, L.
SOURCE, PUB. BY, ETC: U.S. ARMY ELECTRONICS COMMAND
DOCUMENT NUMBER: ECOM-4532
FILED BY (NAME): MOTYKA, P. R.

COMPUTER INTERCONNECTION STRUCTURES: TAXONOMY, CHARACTERISTICS, AND EXAMPLES
DOC. TYPE: PAPER
DOC. DATE: DECEMBER 1975
AUTHORS: ANDERSON, G. A. JENSEN, E. D.
SOURCE, PUB. BY, ETC: ACM; COMPUTING SURVEYS, V 7, #4
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER PRESENTS A TAXONOMY, OR NAMING SCHEME, FOR SYSTEMS OF INTERCONNECTED COMPUTERS. IT IS AN ATTEMPT TO PROVIDE AN IMPLEMENTATION-INDEPENDENT METHOD BY WHICH TO IDENTIFY DESIGNS, AND A COMMON CONTEXT IN WHICH TO DISCUSS THEM. THE TAXONOMY IS BASED ON INTERPROCESSOR MESSAGE HANDLING AND HARDWARE INTERCONNECTION TOPOLOGY, AND DISTINGUISHES TEN BASIC MULTIPLE-COMPUTER ARCHITECTURES. VARIOUS RELEVANT ATTRIBUTES ARE IDENTIFIED AND DISCUSSED,AND EXAMPLES OF ACTUAL DESIGNS ARE GIVEN FOR EACH ARCHITECTURE. KEYWORDS AND PHRASES: DISTRIBUTED PROCESSING, DISTRIBUTED COMPUTERS, MULTIPROCESSORS, MULTICOMPUTERS BUS STRUCTURES, COMPUTER NETWORKS.

COMPUTER PERFORMANCE MEASUREMENT AND EVALUATION METHODS
DOC. TYPE: REPORT
DOC. DATE: JUNE 1974
AUTHORS: SVOBODOVA, L.
SOURCE, PUB. BY, ETC: STANFORD ELECTRONICS LAB, STANFORD UNIVERSITY
DOCUMENT NUMBER: SU-SEL-74-036
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS STUDY CONCENTRATES ON THE MEASUREMENT PROBLEM OF A COMPLEX COMPUTER SYSTEM. SEVERAL ISSUES ARE ATTACKED: SYSTEM REPRESENTATION, EVALUATION AND APPLICATION OF COMPUTER PERFORMANCE EVALUATION TOOLS, POWER OF A PERFORMANCE MONITOR, DESIGN OF A PERFORMANCE MONITOR.

FOR AN EXTERNAL OBSERVER, PERFORMANCE OF A COMPUTER SYSTEM IS THE QUALITY AND THE QUANTITY OF SERVICE DELIVERED BY THE SYSTEM. HOWEVER, A COMPUTER SYSTEM IS A HIERARCHY OF SEVERAL LEVELS, THE LOWEST LEVEL BEING THE CIRCUIT LEVEL, THE HIGHEST THE SOFTWARE SUPPORT LEVEL. PERFORMANCE OF THE SYSTEM AS A WHOLE IS DETERMINED BY PERFORMANCE OF INDIVIDUAL LEVELS. A CONCEPTUAL MODEL OF AN EVALUATED COMPUTER SYSTEM, THE P-MODEL, IS DEFINED IN THIS STUDY USING THE PRINCIPLES OF GENERAL SYSTEMS THEORY; IT PROVIDES A CONVENIENT UNIFORM DESCRIPTION FOR OBSERVING A COMPUTER SYSTEM AT ANY OF THESE LEVELS. THE ELEMENTS OF THE P-MODEL ARE THE LEVEL COMPONENTS: THE OUTPUT ARE PERFORMANCE MEASURES RELEVANT TO THE PARTICULAR LEVEL AND THE PURPOSE OF EVALUATION.

COMPUTER PROGRAM CONFIGURATION ITEM (CPCI) TEST PLANS/PROCEDURES
DOC. TYPE: TECH REPORT
DOC. DATE: MAY 18, 1977
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THIS DATA ITEM DESCRIPTION ESTABLISHES DETAILED QUALIFICATION REQUIREMENTS, CRITERIA, GENERAL METHODS, RESPONSIBILITIES, AND OVERALL PLANNING FOR THE DEVELOPMENT TEST AND EVALUATION (DT&E) QUALIFICATION OF A COMPUTER PROGRAM CONFIGURATION ITEM (CPCI) AND FOR SUBELEMENTS OF THE CPCI.

THE DT&E CPCI TEST PLAN IS NORMALLY OBTAINED IN THE VALIDATION PHASE AS A COMPLETE PLAN APPLICABLE TO THE COMPUTER PROGRAM. PROCEDURES ARE NORMALLY OBTAINED IN THE DESIGN AND DEVELOPMENT PHASES. THIS DATA ITEM DESCRIPTION IS ALSO APPLICABLE TO VALIDATION AND VERIFICATION (V&V) EFFORTS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

COMPUTER SYSTEM MODELING AND SIMULATION
DOC. TYPE: ARTICLE
DOC. DATE: NOVEMBER 1979
AUTHORS: KREUTZER, W.
SOURCE, PUB. BY, ETC: PERFORMANCE EVALUATION REVIEW, VOL 8, NO.1/2
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
TO EVALUATE THE SUITABILITY AND LIMITATIONS OF SOFTWARE FOR COMPUTER SYSTEMS MODELLING, A BASIC COMPREHENSION OF THE STRUCTURE OF SUCH TOOLS MUST BE PROVIDED. A BRIEF DISCUSSION OF CONCEPTUAL REQUIREMENTS FOR THE DESCRIPTION OF DISCRETE MODELS, AND COMPUTER SYSTEM MODELS IN PARTICULAR, IS FOLLOWED BY A SURVEY OF COMMERCIALLY AVAILABLE COMPUTER SIMULATION PACKAGES. SPECIAL AND GENERAL PURPOSE DISCRETE EVENT SIMULATION AND GENERAL PURPOSE PROGRAMMING LANGUAGES ARE ALSO ANALYSED FOR THEIR SUITABILITY FOR THIS CLASS OF APPLICATIONS. THE SURVEY CLOSES WITH SOME RECOMMENDATIONS AND GUIDELINES FOR SELECTION AND APPLICATION OF COMPUTER SYSTEM SIMULATION TOOLS.

TO AID THE ANALYST CONTEMPLATING A COMPUTER SYSTEM MODELLING PROJECT, A BRIEF LIST OF RELEVANT ADDRESSES AND ANNOTATED REFERENCES IS ALSO INCLUDED.

COMPUTER SYSTEM PERFORMANCE ANALYSIS
DOC. TYPE: BOOK
DOC. DATE: 1978
AUTHORS: FERRARI, D.
SOURCE, PUB. BY, ETC: PRENTICE-HALL
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE UNIFYING THEME OF THE BOOK IS A VERY PRAGMATIC ONE: THE EVALUATION STUDY SEEN AS A SET OF PROCEDURES WHOSE END GOAL IS TO GATHER INFORMATION ON THE SYSTEM BEING EVALUATED SO AS TO BE ABLE TO ANSWER CERTAIN PERFORMANCE-RELATED QUESTIONS. THUS, THE BOOK IS MOSTLY CONCERNED WITH THE EVALUATION STUDIES REQUIRED TO SOLVE THE MAIN PROBLEMS WHICH ARISE IN COMPUTER SYSTEMS ENGINEERING AND WITH THE TECHNIQUES TO BE USED IN THESE STUDIES.

COMPUTER SYSTEMS ARCHITECTURE
DOC. TYPE: BOOK
DOC. DATE: 1980
AUTHORS: BAER, JEAN-LOUP
SOURCE, PUB. BY, ETC: COMPUTER SCIENCE PRESS, INC.
DOCUMENT NUMBER: US ISBN 0-914894-15-3 QA76.9.A73B33
FILED BY (NAME): LALA, J. H.
ABSTRACT:
PARTS AND CHAPTERS ARE: PART I, AN OVERVIEW OF THE COMPUTER SPACE. CHAPTERS: 1) HISTORICAL SURVEY OF COMPUTER SYSTEMS ARCHITECTURE, 2) DESCRIPTION OF COMPUTER SYSTEMS; PART II, THE BUILDING BLOCKS AND THEIR INTERACTIONS. CHAPTERS: 3) ARITHMETIC ALGORITHMS, 4) POWERFUL CENTRAL PROCESSORS, 5) THE MEMORY HIERARCHY, 6) MANAGEMENT OF THE MEMORY HIERARCHY, 7) THE CONTROL UNIT AND MICROPROGRAMMING, 8) INPUT-OUTPUT; PART III, COMPLETE SYSTEMS: FROM MICROS TO SUPERCOMPUTERS. CHAPTERS: 9) FROM MICROPROCESSORS TO SUPERCOMPUTERS, 10) SUPERCOMPUTERS, 11) FUTURE TRENDS IN COMPUTER SYSTEMS ARCHITECTURE.

REFERENCES AND EXERCISES ARE PROVIDED AT THE ENDS OF THE CHAPTERS. AN INDEX IS INCLUDED AFTER THE LAST CHAPTER.

CONCURRENCY CONTROL IN DISTRIBUTED DATABASE SYSTEMS
DOC. TYPE: JOURNAL
DOC. DATE: JUNE 1981
AUTHORS: BERNSTEIN, P. A. GOODMAN, N.
SOURCE, PUB. BY, ETC: COMPUTING SURVEYS
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
IN THIS PAPER WE SURVEY, CONSOLIDATE, AND PRESENT THE STATE OF THE ART IN DISTRIBUTED DATABASE CONCURRENCY CONTROL. THE HEART OF OUR ANALYSIS IS A DECOMPOSITION OF THE CONCURRENCY CONTROL PROBLEM INTO TWO MAJOR SUBPROBLEMS: READ-WRITE AND WRITE-WRITE SYNCHRONIZATION. WE DESCRIBE A SERIES OF SYNCHRONIZATION TECHNIQUES FOR SOLVING EACH SUBPROBLEM AND SHOW HOW TO COMBINE THESE TECHNIQUES INTO ALGORITHMS FOR SOLVING THE ENTIRE CONCURRENCY CONTROL PROBLEM. SUCH ALGORITHMS ARE CALLED "CONCURRENCY CONTROL METHODS". WE DESCRIBE 48 PRINCIPAL METHODS, INCLUDING ALL PRACTICAL ALGORITHMS THAT HAVE APPEARED IN THE LITERATURE PLUS SEVERAL NEW ONES. WE CONCENTRATE ON THE STRUCT URE AND CORRECTNESS OF CONCURRENCY CONTROL ALGORITHMS. ISSUES OF PERFORMANCE ARE GIVEN ONLY SECONDARY TREATMENT.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

COST, PERFORMANCE IMPROVE AS FIBER-OPTICS USE GROWS
DOC. TYPE: ARTICLE
DOC. DATE: AUGUST 15, 1983
AUTHORS: HENRY, G.
SOURCE, PUB. BY, ETC: ELECTRONIC ENGINEERING TIMES, 08/15/83
FILED BY (NAME): KEMP, A.
ABSTRACT:
CURRENT STATE OF FIBER OPTICS AND OPTICAL COMPONENTS.

CURRENT ISSUES IN FAULT-TOLERANT COMPUTING
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1, 1983
AUTHORS: HECHT, H.
SOURCE, PUB. BY, ETC: IEEE CPTR SOC 4TH INTL CPT SFTW AND APPLT CONF
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
FAULT TOLERANT SOFTWARE CAN MAKE VALUABLE CONTRIBUTIONS IN THE APPLICATION OF COMPUTERS TO CRITICAL TASKS. THE MOST LIKELY USES OF FAULTTOLERANT SOFTWARE ARE IN FIELDS THAT DEMAND RELIABILITY CURRENTLY NOT OBTAINABLE BY OTHER MEANS, IN PROGRAMS THAT ARE INTENDED TO SERVICE A GREAT VARIETY OF INPUT AND MACHINE STATES, AND THAT NEED TO EXECUTE IN FULL (WITHOUT RECOURSE TO DEFAULT VALUES) DURING COMPUTING CYCLE. FAULT TOLERANCE IS MOST EFFECTIVELY IMPLEMENTED AT A LARGE SCOPE, ENCOMPASSING A FUNCTIONAL MODULE. THERE ARE NO GENERAL GUIDELINES FOR THE VALIDATION OF FAULT TOLERANT SOFTWARE. AT PRESENT, IT IS BEST APPROACHED BY IDENTIFYING SYSTEM STATES THAT ARE UNSAFE, AND THEN DEMONSTRATING THAT THESE CAN NOT BE ENTERED.

DARTS: A TOOL FOR SPECIFICATION AND SIMULATION OF REAL-TIME SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: FURTEK, F. C. DEWOLF, J. BARTON BUCHAN, P. L.
SOURCE, PUB. BY, ETC: PROC. AIAA COMPUTERS IN AEROSPACE III, PP 390-398
FILED BY (NAME): SZULEWSKI, P.

DECOMPOSITION IN RELIABILITY ANALYSIS OF FAULT-TOLERANT SYSTEMS
DOC. TYPE: ARTICLE
AUTHORS: TRIVEDI, K. GEIST, R.
SOURCE, PUB. BY, ETC: IEEE TRANS ON RELIABILITY
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
TWO IMPORTANT PROBLEMS WHICH ARISE IN MODELING FAULT-TOLERANT SYSTEMS WITH ULTRA-HIGH RELIABILITY REQUIREMENTS ARE DISCUSSED. FIRST, ANY ANALYTICAL MODEL OF SUCH A SYSTEM WILL POSSESS A LARGE NUMBER OF STATES, MAKING THE SOLUTION COMPUTATIONALLY INTRACTABLE. THIS LEADS TO THE NEED FOR DECOMPOSITION TECHNIQUES. SECOND, THE COMMON ASSUMPTION OF EXPONENTIAL HOLDING TIMES IN THE STATES IS INTOLERABLE WHILE MODELING SUCH SYSTEMS. APPROACHES TO SOLVING THIS PROBLEM ARE ALSO DISCUSSED.

DESCRIBING SOFTWARE DESIGN IN ADA
DOC. TYPE: ARTICLE
DOC. DATE: SEPTEMBER 1981
AUTHORS: BOOCH, G.
SOURCE, PUB. BY, ETC: SIGPLAN NOT., V 16, #9
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
ADA IS TOOL SUITABLE FOR THE EFFICIENT AND RELIABLE DEVELOPMENT OF SOLUTIONS TO A LARGE PROBLEM DOMAIN. TO COMBAT THE FORTRAN MIND SET AND EXPLOIT THE FULL POWER OF THE LANGUAGE, TOP-DOWN FUNCTIONAL METHODOLOGIES ARE INSUFFICIENT, AND THIS, AN OBJECT-ORIENTED METHODOLOGY IS REQUIRED. THIS PAPER PROVIDES SOME OBSERVATIONS ON SUCH A METHODOLOGY WITH SUBJEQUENT REPRESENTATION OF SOFTWARE DESIGN IN THE ADA PROGRAMMING SUPPORT ENVIRONMENT (APSE).

THE AUTHOR IS WITH THE DEPARTMENT OF ASTRONAUTICS/COMPUTER SCIENCE, USAF ACADEMY, COLORADO 80840 USA.

DESIGN AND ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: 1981
AUTHORS: RAMAMOORTHY, C. V. MA, Y. W.
SOURCE, PUB. BY, ETC: N.-HOLLAND PUB'G CO. NEW YORK USA
DOCUMENT NUMBER: ISBN 0 444 00642 7
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
TO-DAY, DUE TO THE LOW COST OF HARDWARE, THE MOST IMPORTANT PROBLEM IN THE DESIGN OF COMPUTER ARCHITECTURE IS HOW TO CONNECT THE INEXPENSIVE PROCESSORS TOGETHER TO FORM A POWER DISTRIBUTED COMPUTER SYSTEM. THE DESIGN OF COMMUNICATION SYSTEM HAS BECOME MORE DIFFICULT DUE TO THE HIGH COMPUTATIONAL POWER, BUT LOW COMMUNICATION CAPABILITY PROVIDED BY VLSI TECHNOLOGY. IN THIS PAPER, A TOP-DOWN DESIGN METHODOLOGY FOR COMMUNICATION SYSTEMS

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

DESIGN AND ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS * CONTINUED *

IS PRESENTED. ANALYSIS TECHNIQUES THAT ARE USED TO VERIFY THE FUNCTIONAL CORRECTNESS AND TO EVALUATE THE PERFORMANCE OF COMMUNICATION SYSTEMS ARE ALSO ADDRESSED.

THIS PAPER WAS PRESENTED AT AN INVITATIONAL CONFERENCE AND WORKSHOP SPONSORED BY THE IEEE, THE ACM, AND THE INSTITUTE FOR CERTIFICATION OF COMPUTING PROFESSIONALS AND HELD IN ATLANTA, GA USA ON 8-10 SEPTEMBER 1980. THE PROCEEDINGS WERE PUBLISHED BY THE NORTH-HOLLAND PUBLISHING COMPANY IN NEW YORK, USA.

DESIGN OF A NETWORK OPERATING SYSTEM FOR THE DISTRIBUTED DOUBLE-LOOP COMPUTER NETWORK (DDLCN)

DOC. TYPE: PAPER
DOC. DATE: 1982
AUTHORS: LIU, M. T. TSAY, D. P. LIAN, R. C.
SOURCE, PUB. BY, ETC: IFIP 1982, LOCAL COMPUTER NETWORKS, N.-HOLLAND
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER PRESENTS THE FRAMEWORK AND MODEL OF A NETWORK OPERATING SYSTEM (NOS) FOR USE IN DISTRIBUTED SYSTEMS IN GENERAL AND FOR USE IN THE DISTRIBUTED DOUBLE-LOOP COMPUTER NETWORK (DDLCN) IN PARTICULAR. AN INTEGRATED APPROACH IS TAKEN TO DESIGN THE NOS MODEL AND PROTOCOL STRUCTURE. IT IS BASED ON THE OBJECT MODEL AND A NOVEL "TASK" CONCEPT, USING MESSAGE PASSING AS AN UNDERLYING SEMANTIC STRUCTURE. A LAYERED PROTOCOL IS PROVIDED FOR THE DISTRIBUTED SYSTEM KERNEL TO SUPPORT NOS. THIS APPROACH PROVIDES A FLEXIBLE ORGANIZATION IN WHICH SYSTEM-TRANSPARENT RESOURCE SHARING AND DISTRIBUTED COMPUTING CAN EVOLVE IN A MODULAR FASHION. IN THIS PAPER, THE PROTOCOL STRUCTURE IS FIRST PRESENTED. THE NOS MODEL AS WELL AS THE NOTION OF "TASK" IS NEXT DESCRIBED. A TWO-LEVEL PROCESS INTERACTION MODEL IS THEN DISCUSSED. AN INTEGRATED NAMING/PROTECTION SCHEME BASED UPON THE CAPABILITY AND SMALL ACCESS DOMAIN IS DETAILED. FINALLY, THE SYNCHRONIZATION TEMPLATE AND MESSAGE PASSING ARE PROPOSED TO RESOLVE DISTRIBUTED SYNCHRONIZATION PROBLEMS.

DESIGN OF SOFTWARE FOR DISTRIBUTED/MULTIPROCESSOR SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: 1982
AUTHORS: MCKELVEY, T. R. AGRAWAL, D. P.
SOURCE, PUB. BY, ETC: NATL CMPTR CONF 82, SFTWR FOR DSTR MULTIPR SYSS
DOCUMENT NUMBER: TIC# QA76 J74 ENG PEW
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
SOFTWARE DESIGN METHODOLOGIES FOR DISTRIBUTED/MULTIPROCESSOR SYSTEMS ARE INVESTIGATED. PARALLELISM AND MULTITASKING ARE CONSIDERED AS KEY ISSUES IN THE DESIGN PROCESS. PETRI-NETS AND PRECEDENCE GRAPHS ARE PRESENTED AS TECHNIQUES FOR THE MODELING OF A PROBLEM FOR IMPLEMENTATION ON A COMPUTER SYSTEM. TECHNIQUES USING THE PETRI-NET AND PRECEDENCE GRAPH TO DECOMPOSE THE PROBLEM MODEL INTO SUBSETS THAT MAY BE EXECUTED ON A DISTRIBUTED/MULTIPROCESSOR SYSTEM ARE PRESENTED. THESE TECHNIQUES OFFER A SYSTEMATIC DESIGN METHODOLOGY FOR THE DESIGN OF DISTRIBUTED/MULTIPROCESSOR SYSTEM SOFTWARE.

DESIGN OF THE HYBRID AUTOMATED RELIABILITY PREDICTOR

DOC. TYPE: PAPER
AUTHORS: GEIST, R. TRIVEDI, K. DUGAN, J.
SOURCE, PUB. BY, ETC: COMPUTER SCIENCES DEPT, DUKE UNIV
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
WE PRESENT THE DESIGN OF THE HYBRID AUTOMATED RELIABILITY PREDICTOR (HARP) NOW UNDER DEVELOPMENT AT DUKE UNIVERSITY. THE HARP APPROACH TO RELIABILITY PREDICTION IS CHARACTERIZED BY A DECOMPOSITION OF THE OVERALL MODEL INTO FAULT-OCCURRENCE AND FAULT-HANDLING SUB-MODELS. THE FAULT OCCURRENCE MODEL IS A NON-HOMOGENEOUS MARKOV CHAIN WHICH IS SOLVED ANALYTICALLY, WHILE THE FAULT-HANDLING MODEL IS A PETRI NET WHICH IS SIMULATED. HARP PROVIDES AUTOMATED ANALYSIS OF SENSITIVITY TO UNCERTAINTIES IN THE INPUT PARAMETERS AND IN THE INITIAL STATE SPECIFICATIONS. IT THEN PRODUCES A PREDICTED RELIABILITY BAND AS A FUNCTION OF MISS ON TIME, AS WELL AS ESTIMATES OF THE IMPROVEMENT (NARROWING OF THE BAND) TO BE GAINED BY A SPECIFIED AMOUNT OF REDUCTION IN UNCERTAINTY.

DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER. VOLUME I. FTMP PRINCIPLES OF OPERATION

DOC. TYPE: NASA REPORT
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.
SOURCE, PUB. BY, ETC: NASA LANGLEY RESEARCH CENTER (PREP BY CSDL)
DOCUMENT NUMBER: NASA CR-166071 CSDL R-1600
FILED BY (NAME): LALA, J. H.
ABSTRACT:

THIS REPORT IS VOLUME I OF A FOUR-VOLUME REPORT ON THE FAULT-TOLERANT MULTIPROCESSOR (FTMP) PROJECT. IT COVERS IN DETAIL THE FTMP ARCHITECTURE AND PRINCIPLES OF OPERATION, AND IS INTENDED TO SERVE AS A COMPREHENSIVE GUIDE TO THE HARDWARE ORGANIZATION AND OPERATION.

THE FTMP ENGINEERING MODEL WAS CONSTRUCTED BY THE COLLINS AVIONICS GROUP OF THE ROCKWELL

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER. * CONTINUED *

INTERNATIONAL CORPORATION TO THE ARCHITECTURAL AND FUNCTIONAL SPECIFICATIONS PROVIDED BY THE C. S. DRAPER LABORATORY.

THE BASIC ORGANIZATION OF THE FTMP IS THAT OF A GENERAL PURPOSE HOMOGENEOUS MULTIPROCESSOR. THREE PROCESSORS OPERATE ON A SHARED SYSTEM (MEMORY AND I/O) BUS. REPLICATION AND TIGHT SYNCHRONIZATION OF ALL ELEMENTS AND HARDWARE VOTING IS EMPLOYED TO DETECT AND CORRECT ANY SINGLE FAULT. RECONFIGURATION IS THEN EMPLOYED TO "REPAIR" A FAULT. MULTIPLE FAULTS MAY BE TOLERATED AS A SEQUENCE OF SINGLE FAULTS WITH REPAIR BETWEEN FAULT OCCURRENCES.

(NOTE: THIS DOCUMENT HAS BEEN RELEASED BY THE NASA "FOR EARLY DOMESTIC DISSEMINATION"(FEDD). REVIEW FOR GENERAL RELEASE MAY 1985.)

DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER. VOLUME II. FTMP SOFTWARE

DOC. TYPE: NASA REPORT
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.
SOURCE, PUB. BY, ETC: NASA LANGLEY RESEARCH CENTER (PREP BY CSDL)
DOCUMENT NUMBER: NASA CR-166072 CSDL R-1601
FILED BY (NAME): LALA, J. H.
ABSTRACT:

THIS IS VOLUME II OF A FOUR-VOLUME REPORT ON THE FAULT-TOLERANT MULTIPROCESSOR (FTMP) PROJECT. IT COVERS IN DETAIL THE SOFTWARE DEVELOPED FOR THE FTMP.

THE FTMP EXECUTIVE IS A TIMER-INTERRUPT DRIVEN DISPATCHER THAT SCHEDULES ITERATIVE TASKS WHICH RUN AT 3.125, 12.5, AND 25 HZ. MAJOR TASKS WHICH RUN UNDER THE EXECUTIVE INCLUDE SYSTEM CONFIGURATION CONTROL, FLIGHT CONTROL, AND DISPLAY. THE FLIGHT CONTROL TASK INCLUDES AUTOPILOT AND AUTOLAND FUNCTIONS FOR A JET TRANSPORT AIRCRAFT. SYSTEM DISPLAYS INCLUDE STATUS DISPLAYS OF ALL HARDWARE ELEMENTS (PROCESSORS, MEMORIES, I/O PORTS, BUSES), FAILURE LOG DISPLAYS SHOWING TRANSIENT AND HARD FAULTS, AND AN AUTOPILOT DISPLAY.

ALL SOFTWARE IS IN A HIGHER ORDER LANGUAGE (AED, AN ALGOL DERIVATIVE). THE EXECUTIVE IS A FULLY DISTRIBUTED GENERAL PURPOSE EXECUTIVE WHICH AUTOMATICALLY BALANCES THE LOAD AMONG AVAILABLE PROCESSOR TRIADS. PROVISIONS FOR GRACEFUL PERFORMANCE DEGRADATION UNDER PROCESSING OVERLOAD ARE AN INTEGRAL PART OF THE SCHEDULING ALGORITHMS.

(NOTE: THIS DOCUMENT HAS BEEN RELEASED BY THE NASA "FOR EARLY DOMESTIC DISSEMINATION"(FEDD). REVIEW FOR GENERAL RELEASE MAY 1985.)

DEVELOPMENT AND EVALUATION OF A FAULT-TOLERANT MULTIPROCESSOR (FTMP) COMPUTER. VOLUME III. FTMP TEST AND EVALUATION

DOC. TYPE: NASA REPORT
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III LALA, JAYNARAYAN H.
SOURCE, PUB. BY, ETC: NASA LANGLEY RESEARCH CENTER (PREP BY CSDL)
DOCUMENT NUMBER: NASA CR-166073 CSDL R-1602
FILED BY (NAME): LALA, J. H.
ABSTRACT:

THIS REPORT IS VOLUME III OF A FOUR-VOLUME FINAL REPORT ON THE FAULT-TOLERANT MULTIPROCESSOR PROJECT. IT COVERS IN DETAIL THE EXPERIMENTAL TEST AND EVALUATION OF THE FTMP. MAJOR OBJECTIVES OF THIS EXERCISE INCLUDE EXPANDING VALIDATION ENVELOPE, BUILDING CONFIDENCE IN THE SYSTEM, REVEALING ANY WEAKNESSES IN THE ARCHITECTURAL CONCEPTS AND IN THEIR EXECUTION IN HARDWARE AND SOFTWARE, AND IN GENERAL, STRESSING THE HARDWARE AND SOFTWARE.

TO THIS END, PIN-LEVEL FAULTS WERE INJECTED INTO ONE LRU OF THE FTMP AND THE FTMP RESPONSE WAS MEASURED IN TERMS OF FAULT DETECTION, ISOLATION, AND RECOVERY TIMES. A TOTAL OF 21,055 'STUCK-AT-0', 'STUCK-AT-1' AND 'INVERT-SIGNAL' FAULTS WERE INJECTED IN THE CPU, MEMORY, BUS INTERFACE CIRCUITS, BUS GUARDIAN UNITS, AND VOTERS AND ERROR LATCHES. OF THESE, 17,418 WERE DETECTED. AT LEAST 80 PERCENT OF UNDETECTED FAULTS ARE ESTIMATED TO BE ON UNUSED PINS. THE MULTIPROCESSOR IDENTIFIED ALL DETECTED FAULTS CORRECTLY AND RECOVERED SUCCESSFULLY IN EACH CASE. TOTAL RECOVERY TIME FOR ALL FAULTS AVERAGED A LITTLE OVER ONE SECOND. THIS CAN BE REDUCED TO HALF A SECOND BY INCLUDING APPROPRIATE SELF-TESTS.

(NOTE: FEDD STATEMENT, VOLS I&II ABSTRACTS, ALSO APPLIES HERE.)

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

DIGEST OF PAPERS. 12TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING
DOC. TYPE: BOOK
DOC. DATE: JUNE 22, 1982
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC# QA 76.5 .I58 1982
FILED BY (NAME): LALA, J. H.
ABSTRACT:
OF 194 ABSTRACTS SUBMITTED TO THE FTCS-12 COMMITTEE, 150 PAPERS WERE SUBSEQUENTLY SUBMITTED, FROM 16 COUNTRIES. 51 PAPERS, FROM 11 COUNTRIES WERE ACCEPTED FOR PRESENTATION (USA-27; FRANCE-9; JAPAN-4; ALL OTHERS-11). SUBJECTS WERE: ARCHITECTURE(9 PAPERS); RELIABLE SOFTWARE/SOFTWARE NETWORKS/OPERATING SYSTEMS(6); DESIGN FOR TEST/TESTABILITY/FAULT MODELING(15); RELIABILITY MODELING AND EVALUATION(11); AND DIAGNOSIS AND RECOVERY(10). THE OPENING PANEL SESSION FOCUSED ON DEFINING AND REACHING AGREEMENT OF THE FUNDAMENTAL CONCEPTS OF FAULT TOLERANCE, A DIFFICULT AND IMPORTANT TASK WHICH HAD NOT BEEN ACCOMPLISHED PREVIOUSLY. THERE WERE 8 TECHNICAL SESSIONS INCLUDING THE KEYNOTE ADDRESS AND RESPONSES, AND A CLOSING PANEL SESSION DISCUSSING FUTURE DIRECTIONS OF FAULT TOLERANCE AND THE WORK TO BE ADDRESSED IN FUTURE SYMPOSIA. THE NEW DIRECTIONS DISCUSSED WERE PRECIPITATED BY: VLSI TECHNOLOGY; TESTING AND VERIFICATION NEEDS; OPERATING SYSTEM AND DATABASE APPLICATIONS; EXTENSIONS TO COMPUTER-BASED SYSTEMS; AND COMMERCIAL APPLICATIONS. SPONSORED BY IEEE COMPUTER SOCIETY FAULT-TOLERANT TECHNICAL COMMITTEE. 22-24 JUNE 1982. MIRAMAR SHERATON HOTEL, SANTA, MONICA, CA. IEEE CAT.# 82CH1760-8.

DIGEST OF PAPERS: 1ST INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING, PASADENA, CA, MARCH 1971
DOC. TYPE: BOOK
DOC. DATE: 1971
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 10TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING
DOC. TYPE: BOOK
DOC. DATE: 1980
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 11TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING
DOC. TYPE: BOOK
DOC. DATE: 1981
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 13TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING,PALAZZO EX-STELLINE, MILANO, ITALY
DOC. TYPE: BOOK
DOC. DATE: JUNE 28, 1983
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: ISSN#0731-3071, IEEE CAT #83CH1894-5
FILED BY (NAME): LALA, J. H.
ABSTRACT:
LIBRARY OF CONGRESS #79-613257, IEEE COMPUTER SOCIETY ORDER# 477, ISBN #0-8186-0020-9.

DIGEST OF PAPERS: 2ND INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING, NEWTON, MA, JUNE 1972
DOC. TYPE: BOOK
DOC. DATE: 1972
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 3RD INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING, PALO ALTO, CA, JUNE 1973
DOC. TYPE: BOOK
DOC. DATE: 1973
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 4TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING, URBANA, IL, JUNE 1974
DOC. TYPE: BOOK
DOC. DATE: 1974
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

DIGEST OF PAPERS: 5TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING, PARIS, FRANCE, JUNE 1975
DOC. TYPE: BOOK
DOC. DATE: 1975
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 6TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING
DOC. TYPE: BOOK
DOC. DATE: 1976
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 7TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING
DOC. TYPE: BOOK
DOC. DATE: 1977
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 8TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING
DOC. TYPE: BOOK
DOC. DATE: 1978
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DIGEST OF PAPERS: 9TH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING
DOC. TYPE: BOOK
DOC. DATE: 1979
SOURCE, PUB. BY, ETC: IEEE COMPUTER SOCIETY, F-T TECHNICAL COMMITTEE
DOCUMENT NUMBER: TIC#
FILED BY (NAME): LALA, J. H.

DISTRIBUTED AND DECENTRALIZED CONTROL IN FULLY DISTRIBUTED PROCESSING SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: ENSLOW, PHILIP H., JR.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
CERTAINLY ONE OF THE MOST IMPORTANT FACTORS IN DESIGNING AND IMPLEMENTING FULLY DISTRIBUTED PROCESSING SYSTEMS (FDPS) IS THE ISSUE OF DISTRIBUTED AND DECENTRALIZED CONTROL. EXTREMELY LOOSE COUPLING, BOTH PHYSICAL AND LOGICAL, IS AN ESSENTIAL CHARACTERISTIC OF AN FDPS. THIS MODE OF ORGANIZATION AND OPERATION IS QUITE DIFFERENT FROM THE CONTROL OF CENTRALIZED SYSTEMS. THE FIRST STEP IN THE DEVELOPMENT OF DISTRIBUTED AND DECENTRALIZED CONTROL HAS BEEN THE EXAMINATION OF VARIOUS MODELS OF CONTROL THAT MAY PROVIDE THESE FEATURES AND THE OPERATIONAL CHARACTERISTICS OF THOSE MODELS.

DISTRIBUTED DATA PROCESSING --- WHAT IS IT?
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: ENSLOW, PHILIP H., JR.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE, 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
DISTRIBUTED PROCESSING HAS BEEN PRESENTED AS THE MEANS TO OBTAIN IMPROVEMENTS IN A NUMBER OF AREAS OF SYSTEM PERFORMANCE. UTILIZING A LIST OF THESE DESIRED IMPROVEMENTS AS THE MOTIVATIONAL FACTORS, THIS PAPER PRESENTS THE KEY DESIGN CHARACTERISTICS OF SYSTEMS THAT WILL DELIVER A MAJOR PROPORTION OF THESE IMPROVEMENTS. BECAUSE OF THE WIDE USE OF THE TERM "DISTRIBUTED PROCESSING", THE SYSTEMS DESCRIBED HERE ARE IDENTIFIED AS "FULLY DISTRIBUTED".

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

DISTRIBUTED SYSTEM DESIGN DISCIPLINE: AN APPLICATION OF IELL

DOC. TYPE: PAPER
DOC. DATE: 1981
AUTHORS: RHYNE, J. R. DALPEZZO, R. G.
SOURCE, PUB. BY, ETC: DIGEST OF PAPERS, SPRING COMPCON 81
DOCUMENT NUMBER: TIC# QA 76.6 E1; IEEE# CH1626-1/81/0000-0328
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE DISTRIBUTED SYSTEM DESIGN DISCIPLINE IS AN ENVIRONMENT FOR SYSTEM ARCHITECTURE WHICH MAKES USE OF A CHARTLIKE NOTATION. THE METHODOLOGY IS PRESENTED, WITH EXAMPLES OF THE NOTATION, AND ITS ADVANTAGES ARE DISCUSSED. THECOMPONENTS OF A COMPUTER SUPPORT SYSTEM FOR THIS METHODOLOGY ARE DESCRIBED; IN PARTICULAR, WE DESCRIBE A GRAPHICAL ENTRY, EDITING, AND FILING SYSTEM WHICH FACILITATES RECORDING OF DESIGN IN THE NOTATION.

THE AUTHORS ARE WITH THE IBM CORPORATION, FEDERAL SYSTEMS DIVISION (IBM-FSD-MANASSAS).

DMSP FLIGHT LOAD PACKAGE CERTIFICATION FOR FLIGHT 6

DOC. TYPE: VUGRAPHS
DOC. DATE: JUNE 30, 1982
AUTHORS: DALY, KEVIN C.
SOURCE, PUB. BY, ETC: CSDL & TIC
FILED BY (NAME): WERNER, R.E.

DO-178: ITS HISTORY AND PURPOSE

DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: FUNK, D.W.
SOURCE, PUB. BY, ETC: PROC IEEE/AIAA 5TH DIG AV SYS CONF, 10/31-11/03/83
FILED BY (NAME): WERNER, R.
ABSTRACT:
FINALIZED IN NOVEMBER 1981, THE RTCA RELEASED ITS REPORT TITLED "SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION", BETTER KNOWN AS DO-178. THIS DOCUMENT IS HAVING AN IMPACT ON THE AVIONICS INDUSTRY AND IN CERTIFICATION OF SOFTWARE BASED SYSTEMS. THIS PAPER REVIEWS THE HISTORY OF ITS FORMATION. EARLY IN 1980, THE RTCA PERCEIVED A NEED FOR A DOCUMENT DESCRIBING SOFTWARE DEVELOPMENT AND CONTROL TECHNIQUES. SPECIAL COMMITTEE 145 WHICH PREPARED THE DOCUMENT INCLUDED PARTICIPANTS FROM THE FAA, FROM THE AIRFRAME INDUSTRY, FROM THE AVIONICS SUPPLIER INDUSTRIES, FROM THE FEDERAL GOVERNMENT, AND FROM OTHERS INTERESTED IN THE AVIONICS INDUSTRY. THE PAPER REVIEWS SOME OF THE ISSUES FACED BY THE COMMITTEE, INCLUDING SOME THAT WERE ULTIMATELY REJECTED FOR INCLUSION IN THE DOCUMENT. THE PAPER ALSO REVIEWS WHAT IS REQUIRED BY DO-178. INCLUDED IN THIS ARE SOFTWARE DEVELOPMENT AND VERIFICATION PROCEDURES, METHODS, CONFIGURATION MANAGEMENT, QUALITY ASSURANCE, AND DOCUMENTATION.

THE CONFERENCE AT WHICH THIS PAPER WAS PRESENTED WAS HELD IN SEATTLE, WASHINGTON.

ECONOMIC CONSIDERATIONS FOR REAL-TIME NAVAL AIRCRAFT/AVIONIC DISTRIBUTED COMPUTER CONTROL SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: ZEMPOLICH, B. A.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE, 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
USING NAVAL AIRCRAFT/AVIONIC SYSTEMS AS EXAMPLES, ECONOMIC CONSIDERATIONS FOR DISTRIBUTED COMPUTER CONTROL SYSTEMS (DCCS) ARE DISCUSSED. CENTRALIZED, DISTRIBUTED AND FEDERATED PROCESSING ARCHITECTURES ARE USED AS PRIMARY SYSTEMS ALTERNATIVES FROM WHICH TO DEVELOP ECONOMIC FACTORS. TECHNICAL, SCHEDULE AND FINANCIAL RISKS FOR THE SYSTEM ARCHITECTURES ARE PRESENTED. STANDARDIZATION OF COMPUTER HARDWARE AND SOFTWARE IS EXAMINED WITH RESPECT TO ECONOMICS AND OTHER RELATED RISK FACTORS. THE ECONOMIC IMPACT OF SUBSEQUENT LOGISTIC SUPPORT FOR STANDARDIZED VS NONSTANDARD COMPUTER HARDWARE AND SOFTWARE IS IDENTIFIED. SYSTEM CONSIDERATIONS SUCH AS RELIABILITY, MAINTAINABILITY, AVAILABILITY, BUILT-IN-TEST, FAULT TOLERANCE, & REDUNDANCY ARE EXAMINED AS TO RESOURCES AVAILABLE TO DESIGN AND DEVELOP THE DCCS, AND ALSO FROM THE VIEWPOINT OF ECONOMIC IMPACT OF FAILURE OF THE DCCS TO PERFORM AS EXPECTED. THE ECONOMIC IMPACT OF EXTERNAL FACTORS SUCH AS THE RATE OF TECHNOLOGY ADVANCEMENT, TECHNOLOGY INDEPENDENCE, LIMITED PRODUCTION RUNS, AND THE GENERAL LACK OF ECONOMIC LEVERAGE UPON THE MARKET ARE EXAMINED AND RELATED TO THE LIFE-CYCLE SUPPORT REQUIREMENTS OF THE DCCS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

EVALUATION OF MULTIPROCESSOR INTERCONNECT STRUCTURES WITH THE CM* TESTBED
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: WILSON, A. SIEWIOREK, DANIEL P. SEGALL, Z.
SOURCE, PUB. BY, ETC: DEPT ELEC ENGR & COMPUTER SCI DEPT - CMU
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS PAPER PRESENTS A METHOD FOR EMULATING MULTIPROCESSOR ARCHITECTURES ON CM*, A 50 PROCESSOR MULTIPROCESSOR AT CARNEGIE MELLON UNIVERSITY. COMBINED WITH OTHER INSTRUMENTATION TOOLS ALREADY DEVELOPED AT CMU THE RESULT IS A FLEXIBLE TESTBED FOR MULTIPROCESSOR ARCHITECTURE EVALUATION. AN EXPERIMENT TO DEMONSTRATE THE USEFULNESS OF THIS TESTBED IS PRESENTED ALONG WITH RESULTS FOR THREE ARCHITECTURES: RING, NEAREST NEIGHBOR AND FULLY CONNECTED. THESE RESULTS ARE USED TO SHOW HOW THE TESTBED COULD BE USED TO AID IN MULTIPROCESSOR DESIGN. IT IS SHOWN THAT FOR THIS PARTICULAR REAL-TIME APPLICATION A THREE PROCESSOR FULLY CONNECTED STRUCTURE PROVIDES MORE USABLE COMPUTE POWER THAN A SIX PROCESSOR RING.

EXCEPTION HANDLING AND SOFTWARE-FAULT TOLERANCE
DOC. TYPE: JOURNAL REV #: ,
DOC. DATE: OCTOBER 1, 1980
AUTHORS: CRISTIAN, F.
SOURCE, PUB. BY, ETC: 10TH ANN INTL SYMP ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
A GENERAL MODEL ALLOWING A UNIFIED UNDERSTANDING OF PROGRAMMED EXCEPTION HANDLING (FORWARD RECOVERY) AND DEFAULT EXCEPTION HANDLING (BACKWARD RECOVERY) IN SYSTEMS STRUCTURED AS HIERARCHIES OF DATA ABSTRACTIONS IS PRESENTED. THE CAUSE-EFFECT RELATIONSHIP BETWEEN SOFTWARE DESIGN FAULTS AND FAILURE OCCURRENCES IS EXPLORED AND THE ADEQUACY OF BACKWARD RECOVERY IN PROVIDING EFFECTIVE SOFTWARE-FAULT TOLERANCE IS DISCUSSED.

FAULT DETECTION, ISOLATION AND RECONFIGURATION IN FTMP: METHODS AND EXPERIMENTAL RESULTS
DOC. TYPE: PAPER
DOC. DATE: NOVEMBER 2, 1983
AUTHORS: LALA, JAYNARAYAN H.
SOURCE, PUB. BY, ETC: 5TH DIG. AVIONICS SYS'S CONFRNCE, SEATTLE WA
DOCUMENT NUMBER: CSDL P-1661
FILED BY (NAME): LALA, J. H.
ABSTRACT:
THE FAULT-TOLERANT MULTIPROCESSOR (FTMP) IS A HIGHLY RELIABLE COMPUTER DESIGNED TO MEET A GOAL OF (10)-10 FAILURES PER HOUR. TO A LARGE EXTENT THIS LEVEL OF RELIABILITY DEPENDS UPON THE ABILITY TO DETECT AND ISOLATE FAULTS RAPIDLY AND ACCURATELY, PURGE THE FAULTY MODULE, AND DYNAMICALLY RECONFIGURE THE REMAINING GOOD MODULES. THIS PAPER DESCRIBES FAULT DETECTION, ISOLATION AND RECOVERY METHODOLOGY EMPLOYED IN THE FTMP. THE SECOND PART OF THE PAPER DEALS WITH EXPERIMENTAL RESULTS OBTAINED BY ACTUALLY INJECTING FAULTS AT THE PIN LEVEL IN THE FTMP. OVER 21,000 FAULTS WERE INJECTED IN THE CPU, MEMORY, BUS INTERFACE CIRCUITS, AND ERROR DETECTION, MASKING, AND ERROR REPORTING CIRCUITS OF ONE LINE REPLACEABLE UNIT (LRU) OF THE MULTIPROCESSOR. DETECTION, ISOLATION, AND RECONFIGURATION TIMES FOR EACH FAULT WERE RECORDED. THESE RESULTS WERE FOUND TO BE IN CLOSE AGREEMENT WITH EARLIER ASSUMPTIONS MADE IN RELIABILITY MODELING. THE EXPERIMENTAL RESULTS ARE SUMMARIZED IN THIS PAPER.

FAULT TOLERANCE - PRINCIPLES AND PRACTICE
DOC. TYPE: BOOK
DOC. DATE: 1981
AUTHORS: ANDERSON, T. LEE, P. A.
SOURCE, PUB. BY, ETC: PRENTICE/HALL INTERNATIONAL, LONDON
FILED BY (NAME): LALA, J. H.
ABSTRACT:
CHAPTERS ARE: 1) INTRODUCTION; 2)SYSTEM STRUCTURE AND RELIABILITY; 3)FAULT TOLERANCE; 4) FAULT TOLERANT SYSTEMS; 5) ERROR DETECTION; 6) DAMAGE CONFINEMENT AND ASSESSMENT; 7) ERROR RECOVERY; 8) FAULT TREATMENT AND CONTINUED SERVICE; 9) SOFTWARE FAULT TOLERANCE; 10) CONCLUSION.

A LIST OF REFERENCES IS PROVIDED CONTAINING 234 ENTRIES.

AN ANNOTATED BIBLIOGRAPHY IS INCLUDED CONTAINING THREE SECTIONS PROVIDING SUPPLEMENTARY MATERIAL AND REFERENCES.

AN AUTHOR INDEX AND SEPARATE SUBJECT INDEX COMPLETE THE BOOK.

THE BOOK IS AN OUTGROWTH OF A BROAD-BASED RESEARCH PROJECT ON SYSTEM FAULT TOLERANCE SET UP AT THE UNIVERSITY OF NEWCASTLE UPON TYNE IN THE EARLY 1970'S. THE AUTHORS, WHO ARE LEADING MEMBERS OF THE RESEARCH PROJECT, HAVE ACHIEVED AN ADMIRABLE SYSTHESIS OF THE MAJOR HARDWARE AND SOFTWARE ISSUES IN, AND OF THE BEST CURRENT PRACTICE CONCERNING, THE DESIGN OF FAULT-TOLERANT COMPUTING SYSTEMS. THEIR TEXT INCLUDES A CAREFULLY BALANCED YET VERY READABLE DISCUSSION OF BASIC IDEAS AND CONCEPTS, AND OF THE TECHNICAL DETAILS OF A NUMBER

FAULT TOLERANCE - PRINCIPLES AND PRACTICE * CONTINUED *
OF IMPORTANT CONTEMPORARY FAULT-TOLERANT COMPUTING SYSTEMS.

FAULT TOLERANT PROCESSOR CONCEPTS AND OPERATION
DOC. TYPE: PAPER
DOC. DATE: MAY 1983
AUTHORS: SMITH, T. BASIL, III
SOURCE, PUB. BY, ETC: (SUB'D TO FTCS #14, FLORIDA, JUNE 84)
DOCUMENT NUMBER: CSDL P-1727
FILED BY (NAME): LALA, J. H.
ABSTRACT:
THE FTP CONCEPT IS AN ATTEMPT TO CREATE, AND EXTEND TO THE F-T COMPUTING ENVIRONMENT, THE SAME GENERAL PURPOSE CHARACTERISTICS WHICH HAVE PROVED TO SUCCESSFUL FOR COMPUTING IN GENERAL. IT MIMICS THE ARCHITECTURE AND OPERATION OF MODERN SIMPLEX MACHINES, THEREBY CREATING AN ABILITY TO GROW WITH AND BUILD ON THE GENERAL PURPOSE TECHNICAL BASE, A BASE WHICH CAN BE INCREMENTALLY ENHANCED. IN DOING SO IT ALSO CREATES A SITUATION IN WHICH MUCH OF THE EXISTING SIMPLEX GENERAL PURPOSE COMPUTING BASE CAN BE TAPPED. THE EXPERIENCE TO DATE, PARTICULARLY WITH THE MORE RECENT PROTOTYPES, INDICATES THAT THE FAULT TOLERANT PROCESSOR CONCEPT IS SOUND, AND THAT SUCH AN ARCHITECTURE CAN BE CREATED AND EVOLVED TO FILL THE NEED FOR FAULT TOLERANT COMPUTING AT REASONABLE PROGRAM AND PROJECT COSTS.

FAULT-TOLERANT COMPUTING
DOC. TYPE: ARTICLE
DOC. DATE: JULY 1, 1979
AUTHORS: HECHT, H.
SOURCE, PUB. BY, ETC: IEEE TRANSACTIONS ON RELIABILITY
FILED BY (NAME): FURTEK, F. C.

FAULT-TOLERANT SOFTWARE
DOC. TYPE: ARTICLE
DOC. DATE: AUGUST 1979
AUTHORS: HECHT, HERBERT
SOURCE, PUB. BY, ETC: IEEE TR ON RELIABILITY, V R-28, #3
FILED BY (NAME): FURTEK, F. C.

FORMAL SPECIFICATION AND MECHANICAL VERIFICATION OF SIFT: A FAULT-TOLERANT FLIGHT CONTROL SYSTEM
DOC. TYPE: ARTICLE
DOC. DATE: JULY 1982
AUTHORS: MELLIAR-SMITH, P. SCHWARTZ, R.
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
THIS PAPER DESCRIBES THE FORMAL SPECIFICATION AND PROOF METHODOLOGY EMPLOYED TO DEMONSTRATE THAT THE SIFT COMPUTER MEETS ITS REQUIREMENTS. THE HIERARCHY OF DESIGN SPECIFICATIONS IS SHOWN, FROM VERY ABSTRACT DESCRIPTIONS OF SYSTEM FUNCTION DOWN TO THE IMPLEMENTATION. THE MOST ABSTRACT DESIGN SPECIFICATIONS ARE SIMPLE AND EASY TO UNDERSTAND, ALMOST ALL DETAILS OF THE REALIZATION HAVING BEEN ABSTRACTED OUT, AND CAN BE USED TO ENSURE THAT THE SYSTEM FUNCTIONS RELIABLY AND AS INTENDED. A SUCCESSION OF LOWER LEVEL SPECIFICATIONS REFINE THESE SPECIFICATIONS INTO MORE DETAILED AND MORE COMPLEX VIEWS OF THE SYSTEM DESIGN, CULMINATING IN THE PASCAL IMPLEMENTATION. THE PAPER DESCRIBES THE RIGOROUS MECHANICAL PROOF THAT THE ABSTRACT SPECIFICATIONS ARE SATISFIED BY THE ACTUAL IMPLEMENTATION.

FOUNDATIONS OF THE NEXT-GENERATION MICROPROCESSORS
DOC. TYPE: PAPER
DOC. DATE: 1981
AUTHORS: BOUTE, R. T.
SOURCE, PUB. BY, ETC: EUROMICRO 1981; NORTH-HOLLAND PUB'G CO., AMST'DM
DOCUMENT NUMBER: ISBN 0 444 86282 X
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE FORTHCOMING GENERATION OF MICROPROCESSORS USHERS IN A RADICAL DEPARTURE FROM THE TRADITIONAL APPROACH IN MICROCOMPUTER ARCHITECTURE AND, MORE IMPORTANTLY, IN PROGRAMMING. RATHER THAN INVOLVING A MERE EXTRAPOLATION IN WORD LENGTH (E.G. 32 OR 64 BITS), THIS NEW GENERATION WILL BE SHAPED BY CONCEPTS ARISING FROM COMPUTER SCIENCE, MAINLY PROGRAMMING LANGUAGE AND OPERATING SYSTEM THEORY. THE PURPOSE IS TO CREATE A HIGH-LEVEL, CONCEPTUAL ENVIRONMENT FOR PROBLEM SOLVING, THUS CONTRIBUTING TO DESIGN QUALITY AND EFFECTIVENESS.

ALTHOUGH THE PRESENTATION EMPHASIZES GENERAL PRINCIPLES, IT WILL BE STRONGLY ORIENTED TOWARDS THE INTEL IAPX432, WHICH IS THE FORERUNNER OF THESE NEW DEVICES. ... THE PRESENTATION IS ORIENTED TOWARDS READERS WITH ONLY A MINIMAL ACQUAINTANCE WITH MODERN HIGHER-ORDER LANGUAGES, BUT OCCASIONALLY ALSO INTRODUCES SOME MORE ADVANCED VIEWS.

THE AUTHOR IS WITH THE BELL TELEPHONE MANUFACTURING COMPANY, ANTWERPEN, BELGIUM.

IMPLEMENTING FUNCTIONS: MICROPROCESSORS AND FIRMWARE. 7TH EUROMICRO SYMP. ON MICROPROCESSING & MICROPROGRAMMING, PROC PP 271-87, 8-10 SEPT. 1981, PARIS, FRANCE.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

FTMP - A HIGHLY RELIABLE FAULT-TOLERANT MULTIPROCESSOR FOR AIRCRAFT
DOC. TYPE: ARTICLE
DOC. DATE: OCTOBER 1978
AUTHORS: HOPKINS, ALBERT L. SMITH, T. BASIL, III LALA, JAYNARAYAN H.
SOURCE, PUB. BY, ETC: PROCEEDINGS IEEE, V 66, #10, PP 1221-1239
FILED BY (NAME): LALA, J. H.
ABSTRACT:
FTMP IS A DIGITAL COMPUTER ARCHITECTURE WHICH HAS EVOLVED OVER A TEN-YEAR PERIOD IN CONNECTION WITH SEVERAL LIFE-CRITICAL AEROSPACE APPLICATIONS. MOST RECENTLY IT HAS BEEN PROPOSED AS A FAULT-TOLERANT CENTRAL COMPUTER FOR CIVIL TRANSPORT AIRCRAFT APPLICATIONS. A WORKING EMULATION HAS BEEN OPERATING FOR SOME TIME, AND THE FIRST ENGINEERING PROTOTYPE IS SCHEDULED TO BE COMPLETED IN LATE 1979. FTMP IS DESIGNED TO HAVE A FAILURE RATE DUE TO RANDOM CAUSES OF THE ORDER OF (10)-10 FAILURES PER HOUR, ON TEN-HOUR FLIGHTS WHERE NO AIRBORNE MAINTENANCE IS AVAILABLE. THE PREFERRED MAINTENANCE INTERVAL IS OF THE ORDER OF HUNDREDS OF FLIGHT HOURS, AND THE PROBABILITY THAT MAINTENANCE WILL BE REQUIRED EARLIER THAN THE PREFERRED INTERVAL IS DESIRED TO BE AT MOST A FEW PERCENT. ... FAILURE-RATE MODELS AND NEMERICAL RESULTS ARE DESCRIBED FOR BOTH PERMANENT AND INTERMITTENT FAULTS. A DISPATCH PROBABILITY MODEL IS ALSO PRESENTED. EXPERIENCE WITH AN EXPERIMENTAL EMULATION IS DESCRIBED.

FUNCTIONAL DOCUMENTATION - A PRACTICAL AID TO THE ORDERLY SOLUTION OF THE SYSTEM DESIGN PROBLEM
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: MARTIN, J. T.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
TIC #AD-A109-274. THE AUTHOR IS WITH FERRANTI COMPUTER SYSTEMS LIMITED, WESTERN ROAD, BRACKNELL, BERKSHIRE, ENGLAND.

FUNCTIONAL PROGRAM TESTING
DOC. TYPE: ARTICLE
DOC. DATE: MARCH 1980
AUTHORS: HOWDEN, W.
SOURCE, PUB. BY, ETC: IEEE TRANSACTIONS ON SOFTWARE ENG. V. SE-6 #2
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
AN APPROACH TO FUNCTIONAL TESTING IS DESCRIBED IN WHICH THE DESIGN OF A PROGRAM IS VIEWED AS AN INTEGRATED COLLECTION OF FUNCTIONS. THE SELECTION OF TEST DATA DEPENDS ON THE FUNCTIONS USED IN THE DESIGN AND ON THE VALUE SPACES OVER WHICH THE FUNCTIONS ARE DEFINED. THE BASIC IDEAS IN THE METHOD WERE DEVELOPED DURING THE STUDY OF A COLLECTION OF SCIENTIFIC PROGRAMS CONTAINING ERRORS. THE METHOD WAS THE MOST RELIABLE TESTING TECHNIQUE FOR DISCOVERING THE ERRORS. IT WAS FOUND TO BE SIGNIFICANTLY MORE RELIABLE THAN STRUCTURAL TESTING. THE TWO TECHNIQUES ARE COMPARED AND THEIR RELATIVE ADVANTAGES AND LIMITATIONS ARE DISCUSSED.

FUNCTIONAL VERSUS COMMUNICATION STRUCTURES IN MODERN AVIONIC SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: BRAMMER, K. WEIMANN, A.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE, 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
IMPLICATIONS OF INCREASED FUNCTIONAL AND COMMUNICATION INTERFACES ON AVIONIC SYSTEM STRUCTURES ARE ANALYZED (ESPECIALLY THE PASSAGE FROM FUNCTIONAL DESIGN TO IMPLEMENTED COMMUNICATION STRUCTURE OF THE AIRBORNE ELECTRONIC SYSTEM). THE DISTRIBUTED ORGANISATION OF AN AVIONIC SYSTEM, REALISATION OF WHICH IS GREATLY SIMPLIFIED BY BUS TYPE INTRASYSTEM SIGNAL TRANSMISSION, IS COMPARED TO CONVENTIONAL HIERARCHICAL SYSTEM ORGANSATION. PROS AND CONS OF BOTH ORGANISATIONS ARE REVIEWED ESPECIALLY WITH RESPECT TO INTERFACE EFFICIENCY, CABLING REQUIREMENTS AND THE TYPICAL TOPOLOGY OF AVIONIC SYSTEMS.

THE TOPIC IS ILLUSTRATED BY STRUCTURES OF A CONVENTIONAL AND A MODERN AVIONIC SYSTEM.

IT IS SUGGESTED THAT FOR AVIONIC SYSTEMS THE CONFLICTING GOALS OF DETERMINISTIC SYSTEM BEHAVIOUR (REQUIRING FEW FUNCTIONAL AND COMMUNICATION INTERFACES AND TIGHT CONTROL) AND ENHANCED AVAILABILITY (REQUIRING DISTRIBUTION OF RESOURCES, REALLOCATION OF FUNCTIONS AND MANY COMMUNICATION INTERFACES) REQUIRE MORE RESEARCH AND PRACTICAL EXPERIENCE IN ORDER TO HARMONIZE THEM AND TO ESTABLISH NEW ADEQUATE AND GENERALLY ACCEPTED AVIONIC SYSTEM IMPLEMENTATION PROCEDURES.

THE AUTHORS ARE CITIZENS OF WEST GERMANY.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

FURTHER DEVELOPMENTS ON THE CAMBRIDGE RING NETWORK AT THE UNIVERSITY OF KENT
DOC. TYPE: PAPER
DOC. DATE: 1982
AUTHORS: BINNS, S. E. DALLAS, I. N. SPRATT, E. B.
SOURCE, PUB. BY, ETC: IFIP, 1982, LOCAL COMPUTER NETWORKS, N.-HOLLAND
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER GIVES AN OVERVIEW OF THE WORK CARRIED OUT ON THE CAMBRIDGE RING LOCAL AREA NETWORK BY THE COMPUTING LABORATORY AT THE UNIVERSITY OF KENT, CONCENTRATING ON THE PERIOD FROM XMAS 1979 TO JUNE 1981. FUTURE PLANS ARE ALSO DISCUSSED. IT IS IN PART BASED ON THE MATERIAL IN (20). A GENERAL INTRODUCTION TO THE SUBJECT OF LOCAL AREA NETWORKS IS GIVEN IN (05), AND AN EXTENSIVE BIBLIOGRAPHY IS GIVEN IN (16). THE READER WHO IS UNFAMILIAR WITH CAMBRIDGE RINGS WILL FIND A BRIEF DESCRIPTION IN APPENDIX 1. MORE DETAILED ACCOUNTS ARE GIVEN IN (10) AND (22). APPENDIX 2 CONTAINS INFORMATION ON RING PROTOCOLS. REFERENCES (12) AND (15) CONTAIN INFORMATION ON LOCAL AREA NETWORK ISSUES IN THE CONTEXT OF UK UNIVERSITITIES AND US UNIVERSITIES RESPECTIVELY.

GENERALIZED COMPUTER SYSTEMS SIMULATOR II (GCSS II) USER'S GUIDE
DOC. TYPE: REPORT
AUTHORS: HART, L.
SOURCE, PUB. BY, ETC: SOFTWARE AND COMPUTER DIRECTORATE N.A.D.C.
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE OBJECTIVE OF GCSS II IS TO PROVIDE A FACILITY FOR THE EVALUATION OF COMPUTER SYSTEM ARCHITECTURES BY MEASURING HARDWARE UTILIZATION AS A FUNCTION OF THE APPLICATION SOFTWARE AND THE HARDWARE CHARACTERISTICS SUCH AS PROCESSOR CLOCK TIME AND INSTRUCTION REPETOIRE, MEMORY SIZE AND ACCESS SPEEDS, BUS SPEEDS AND INTERCONNECTIONS. ALTHOUGH THE SIMULATOR COULD BE USED TO EVALUATE ANY KIND OF COMPUTER SYSTEM, IT WAS PRIMARILY DESIGNED WITH THE IDEA OF EVALUATING MULTI-PROCESSOR, MULTI BUS ARCHITECTURE SYSTEMS.

GCSS II WAS DEVELOPED TO BE THE EVALUATION TOOL WITHIN A COMPUTER SYSTEM ARCHITECTURE DESIGN AND EVALUATION FACILITY. ALTHOUGH NOT A SYSTEM DESIGN TOOL ITSELF, THE ANALYSIS OF THE OUTPUTS CAN BE USED TO UNCOVER POSSIBLE DESIGN WEAKNESSES AND RECOMMEND SYSTEM IMPROVEMENTS EARLY IN THE DESIGN PHASE. GCSS II CAN BE USED AS A TOOL FOR EVALUATING:

(A) PROCESSOR LOADING AND TASK ALLOCATION

(B) PROCESSING STRUCTURE ALTERNATIVES

(C) DATA MANAGEMENT TRADEOFFS

(D) BUS PROTOCOL AND STRUCTURE TRADEOFFS.

GENERALIZED POLLING ALGORITHMS FOR DISTRIBUTED SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: WOLF, J. K.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
TIC #AD-A109-274. THE AUTHOR IS WITH THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, UNIVERSITY OF MASSACHUSETTS, AMHERST, MA 01003 USA.

HIPO AND INTEGRATED PROGRAM DESIGN
DOC. TYPE: ARTICLE
DOC. DATE: 1976
AUTHORS: STAY, J. F.
SOURCE, PUB. BY, ETC: IBM SYSTEM JOURNAL, V 15, #2, PP 143-154 (HIPO)
FILED BY (NAME): SZULEWSKI, P.

HMOS PROCESSING IMPROVES RADIATION RESISTANCE OF VLSI MEMORIES AND MICROPROCESSORS
DOC. TYPE: PAPER
DOC. DATE: NOVEMBER 30, 1982
AUTHORS: DAVIS WOODS ET AL
SOURCE, PUB. BY, ETC: GOVT. MICROCIRCUITS APPLICATIONS CONF., 11/82
FILED BY (NAME): KEMP, A.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

IMPLEMENTATION AND PERFORMANCE EVALUATION OF COMPUTER FAMILIES
DOC. TYPE: ARTICLE
DOC. DATE: JUNE 1981
AUTHORS: SNOW, E. SIEWIOREK, DANIEL P.
SOURCE, PUB. BY, ETC: IEEE TRANS ON COMNPUTERS, VOL C-30, NO.6
DOCUMENT NUMBER: 0018-9340/81/0600-0443
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS CORRESPONDENCE PROPOSES A MODEL USING MICROCYCLE AND MEMORY READ PAUSE TIMES TO ACCOUNT FOR VARIATION IN PERFORMANCE BETWEEN MEMBERS OF A COMPUTER FAMILY. WHEN APPLIED TO THE DEC PDP-11 AND IBM S/360-S/370 FAMILIES, THE MODEL EXPLAINS OVER 90 PERCENT OF THE VARIATION. THIS MODEL IS USEFUL FOR INITIAL FAMILY PLANNING, AS WELL AS DESIGN OF INDIVIDUAL FAMILY MEMBERS.

IMPLEMENTATION LANGUAGES FOR DATA ABSTRACTIONS
DOC. TYPE: PAPER
DOC. DATE: MARCH 19, 1984
AUTHORS: LEACH, DANIEL M. SATKO, JAMES E.
SOURCE, PUB. BY, ETC: PHOENIX CONF. COMPTRS & CMMNICATNS (TBPUBLISHED)
FILED BY (NAME): KNOSP, A. A.
ABSTRACT:
IN THIS PAPER, WE COMPARE VARIOUS LANGUAGES AS POTENTIAL IMPLEMENTATION LANGUAGES FOR DATA ABSTRACTIONS. BOTH SIMPLE ABSTRACT DATA TYPES (SUCH AS STACKS, QUEUES, LINKED LISTS, TREES, AND GRAPHS) AS WELL AS MORE ADVANCED ABSTRACT DATA TYPES COMMONLY FOUND IN THE WORK PLACE (SUCH AS SCREENS, DOCUMENTS, AND TEXT FILES) ARE CONSIDERED ALONG WITH CONCRETE FUNCTIONS FOR MANIPULATING THESE ABSTRACT TYPES.

PAPER TO BE PRESENTED AT THE PHOENIX CONFERENCE ON COMPUTERS AND COMMUNICATIONS MARCH 19 THROUGH 21, 1984.

IMPLEMENTATION OF A REAL-TIME DISTRIBUTED COMPUTER SYSTEM IN ADA
DOC. TYPE: ARTICLE
DOC. DATE: 1983
AUTHORS: LANE, D. S. HULING, G. BARDIN, B. M.
SOURCE, PUB. BY, ETC: AIAA
DOCUMENT NUMBER: PAPER NO. 83-2407
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER REPORTS ON A PROTOTYPE REAL-TIME DISTRIBUTED COMPUTER SYSTEM WHICH WILL BE IMPLEMENTED IN ADA (ADANET). THE GOALS OF THE ADANET SYSTEM ARE TO SUPPORT TRANSPARENT DISTRIBUTION OF APPLICATION SOFTWARE AND INCREMENTAL GROWTH, AND TO PROVIDE FAULT TOLERANT CAPABILITIES. SOME OF THE MOTIVATIONS ANDMETHODS FOR DISTRIBUTING SOFTWARE IN A LOCAL NETWORK OF COMMUNICATING PROCESSORS ARE DISCUSSED, IN ADDITION TO THE HARDWARE CONFIGURATION AND SOFTWARE DEVELOPMENT FACILITIES. THE MODEL OF DISTRIBUTED ADA PROGRAMS IS THEN DESCRIBED. AFTER THE PROTOTYPE SOFTWARE IS IMPLEMENTED, THE PROJECT WILL FOCUS ON ASSESING THE PERFORMANCE CHARACTERISTICS OF ADANET, SPECIFICALLY, THE DISTRIBUTION, EXECUTIVE SOFTWARE, AND ADA LANGUAGE OVERHEADS.

THE AUTHORS ARE WITH THE HUGHES AIRCRAFT COMPANY.

IMPLEMENTATION OF DO-178 IN COMMERCIAL AIRBORNE SOFTWARE DESIGN
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: UCZEKAJ, J.S.
SOURCE, PUB. BY, ETC: PROC IEEE/AIAA 5TH DIG AV SYS CONF, 10/31-11/03/83
FILED BY (NAME): WERNER, R.
ABSTRACT:
RTCA DOCUMENT DO-178 ESTABLISHES SOFTWARE DEVELOPMENT AND TESTING REQUIREMENTS TO REDUCE THE PROBABILITY OF AN ERROR RESIDING IN AN AIRBORNE SOFTWARE PROGRAM THAT COULD PROVIDE INACCURATE OR MIDLEADING INFORMATION, TO A LEVEL EQUIVALENT TO THE PROBABILITIES APPLIED TO ANALOG SYSTEMS. HOWEVER, CURRENT SOFTWARE TECHNOLOGY DOES NOT IDENTIFY A METHOD OF CALCULATING A SIMILAR NUMBER FOR A SOFTWARE PROGRAM. THEREFORE, DO-178 IDENTIFIES SEQUENCES OF EVENTS THAT MUST BE ACCOMPLISHED ALONG WITH ADEQUATE VERIFICATION AND VALIDATION ACTIVITIES THAT MUST OCCUR IN ORDER TO PROVIDE A "LEVEL OF CONFIDENCE" IN THE SOFTWARE PROGRAM.

THE CURRENT APPLICATION OF DO-178 IS ESTABLISHING PRECEDENTS REGARDING THE REQUIREMENTS FOR AIRBORNE SOFTWARE. IN ADDITION, PRECEDENTS ARE BEING ESTABLISHED FOR THE REQUIREMENTS OF POST-CERTIFICATION MODIFICATIONS TO NON-ESSENTIAL, ESSENTIAL AND FLIGHT CRITICAL AIRBORNE SOFTWARE.

THIS PAPER WILL EXAMINE THE REQUIREMENTS AND INTENT OF DO-178 AS APPLIED TO COMMERCIAL AIRBORNE SOFTWARE.

THE CONFERENCE AT WHICH THIS PAPER WAS PRESENTED WAS HELD IN SEATTLE, WASHINGTON.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

INDEPENDENT VERIFICATION AND VALIDATION PROGRAM DEVELOPMENT
DOC. TYPE: REPORT
DOC. DATE: FEBRUARY 15, 1980
AUTHORS: DALY, KEVIN C.
SOURCE, PUB. BY, ETC: TIC
FILED BY (NAME): WERNER, R.E.

INTEGRATED AIRFRAME/PROPULSION CONTROL SYSTEM ARCHITECTURES (IAPSA) STUDY
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: STERN, A. D.
SOURCE, PUB. BY, ETC: AIAA
DOCUMENT NUMBER: PAPER NO. 83-2158
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE PURPOSE OF THE IAPSA STUDY WAS TO DEFINE, EVALUATE, AND SELECT CANDIDATE AIRCRAFT CONTROL SYSTEM (ACS) ARCHITECTURES FOR 1990'S HIGH PERFORMANCE AIRPLANES. AN AIRPLANE POSSESSING A HIGH DEGREE OF AIRFRAME/PROPULSION COUPLING WAS SELECTED AS THE APPLICATION AIRCRAFT AND ACS ARCHITECTURES WERE DEVELOPED USING TWO APPROACHES. THE FIRST APPROACH DEVELOPED A BASELINE STATE-OF-THE-ART SYSTEM THAT PRESERVED MUCH OF THE AUTONOMY OF TRADITIONAL FLIGHT AND PROPULSION CONTROL SUBSYSTEMS. THE SECOND WAS A TOP-DOWN "INTEGRATION FROM THE START" APPROACH WHICH WAS NOT CONSTRAINED BY TRADITION AND WHICH EMPLOYED SUITABLE EMERGING TECHNOLOGIES.

THE SELECTED ARCHITECTURE, CALLED "CENTRAL/DIRECT", HAS THE FOLLOWING FEATURES: 1) A HIGHLY REDUNDANT CENTRALIZED COMPUTER COMPLEX IN WHICH ALL CONTROL PROCESSING IS PERFORMED; AND 2) ALL ELECTRONICS RESIDE IN TWO SIMILAR (FAIL-OPERATIVE) DISPERSED BOXES THAT ARE OPTICALLY CONNECTED IN A DIRECT MANNER TO SENSORS AND SERVOVALVES.

THE STUDY IS THE USAF/NAVY/GENERAL DYNAMICS/BENDIX ADVANCED FIGHTER TECHNOLOGY INTEGRATION USING THE F-16 AS A DEMONSTRATOR.

THE AUTHOR IS WITH THE BOEING MILITARY AIRPLANE CO., P.O.BX 3707, M/S 41-23, SEATTLE, WA 98124.

INTEGRATED VERIFICATION AND TESTING SYSTEM (IVTS) FOR HAL/S PROGRAMS
DOC. TYPE: PAPER
DOC. DATE: JULY 1983
AUTHORS: SENN, E. H. AMES, K. R. SMITH, K. A.
SOURCE, PUB. BY, ETC: PROCEEDINGS OF SOFTFAIR 1983
FILED BY (NAME): SZULEWSKI, P.

INTEL TO BOLSTER 432 SALES WITH FAULT-TOLERANT COMPONENTS
DOC. TYPE: ARTICLE
DOC. DATE: MARCH 1983
AUTHORS: STREHLO, K.
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
INTEL CORP. HAS HAD LITTLE MARKET SUCCESS IN PUSHING ITS TWO-YEAR-OLD IAPX 432 32 BIT MICROMAINFRAME CHIP SET BY TOUTING ITS ARCHITECTURAL PROTECTION AGAINST FAULTY SOFTWARE. BUT THE COMPANY HOPES THE ADDITION OF TOLERANCE FOR HARDWARE FAULTS WILL MAKE THE 432 A MARKET WINNER, ALTHOUGH INITIAL SYSTEMS WILL BE RELATIVELY EXPENSIVE.

TWO NEW INTEL VLSI COMPONENTS, THE 43204 BUS INTERFACE UNIT AND THE 43205 MEMORY CONTROL UNIT, INCORPORATE THE DETECTION AND RECOVERY LOGIC REQUIRED TO ADD A RANGE OF FAULT-TOLERANT CAPABILITIES TO A 432-BASED SYSTEM. THE SAME CHIPS SUPPORT AN INTERCONNECTION ARCHITECTURE THAT ALLOWS THE MODULAR AND TRANSPARENT EXTENSION OF THE PROCESSING POWER AND BUS BANDWIDTH OF SUCH A SYSTEM.

GORDON REID, MARKETING MANAGER AT INTEL'S SPECIAL SYSTEMS OPERATION, ALOHA, ORE., EXPECTS THIS CONFIGURATION FLEXIBILITY AND RELIABILITY, PLUS A FIVEFOLD PERFORMANCE IMPROVEMENT SINCE THE 432 WAS INTRODUCED, WILL OVERCOME MARKET RESISTANCE TO THE 432 BEFORE 1985, WHEN THERE WILL BE A $19-BILLION DEMAND FOR HIGH-AVAILABILITY MACHINES, ACCORDING TO DATAQUEST, INC., A CUPERTINO, CALIF. MARKER RESEARCH FIRM.

IS ADA THE ANSWER?
DOC. TYPE: REPORT
AUTHORS: BULMAN, DAVID M.
FILED BY (NAME): KNOSP, A.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

ISSUES IN SOFTWARE RELIABILITY
DOC. TYPE: PAPER
DOC. DATE: JULY 1981
AUTHORS: RAMAMOORTHY, C. V. GANESH, S. L.
SOURCE, PUB. BY, ETC: IEEE; PROC, SYMP, REL IN DISTR SFTWR & DTABAS SYSS
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
IT IS IMPORTANT TO ENSURE THAT COMPUTER SYSTEMS FOR CRITICAL REAL-TIME APPLICATIONS ARE SUFFICIENTLY RELIABLE. THIS REQUIREMENT ENCOMPASSES THE NEED BOTH TO ENSURE THE CORRECTNESS OF THE DESIGN OF THE COMBINED HARDWARE-SOFTWARE SYSTEM BEFORE IT IS PUT INTO OPERATION AND TO SECURE THE SYSTEM FROM DETERIORATION IN THE OPERATIONAL PHASE AS THEY SYSTEM IS PATCHED AND AUGMENTED AND HARDWARE PARTS WEAR OUT. THE COMPLEXITY OF MANY SOFTWARE SYSTEMS MAKES THE FULFILMENT OF THESE REQUIREMENTS ONEROUS. FORMAL PROOFS OF CORRECTNESS FOR SOFTWARE ARE USUALLY LENGTHY AND NOT COMPLETELY CONVINCING. THEREFORE TESTING PROCEDURES AND RELIABILITY MODELS ARE REQUIRED. WE INTRODUCE AND CLASSIFY THE MODELS THAT HAVE BEEN PROPOSED IN THE LITERATURE. WE ALSO DISCUSS METHODS FOR COMPARING THE ADEQUACY OF THE TESTING METHODS USED. THE NEED FOR RESEARCH ON INTEGRATED HARDWARE-SOFTWARE RELIABILITY MODELS IS DISCUSSED. SUCH MODELS WILL BE REQUIRED IN ORDER TO DERIVE GOOD RELIABILITY ESTIMATES OF SYSTEMS IN WHICH REDUNDANCY OF HARDWARE AND SOFTWARE IS EXPLOITED FOR FAULT-TOLERANCE E.G. DISTRIBUTED SYSTEMS.

THE AUTHORS ARE WITH THE UNIV OF CALIFORNIA, BERKELEY, CA 94720 USA.

LESSONS LEARNED IN THE DEVELOPMENT OF THE 757/767 FLIGHT MANAGEMENT SYSTEM EQUIPMENT
DOC. TYPE: PAPER
AUTHORS: SUTCLIFFE, P.
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THE BOEING 757/767 FLIGHT MANAGEMENT SYSTEM IS COMMON ON BOTH THE 757 AND 767. IT INCLUDES THE BASIC ELECTRONIC FLIGHT DECK DISPLAYS, AUTOFLIGHT SYSTEMS INCLUDING SENSORS, AIR DATA AND INERTIAL REFERENCE SYSTEMS, AND THE ENGINE INSTRUMENTS AND CREW ALERTING SYSTEMS. ALL SYSTEMS ARE DIGITAL, WITH INTERCONNECTIONS BY THE ARINC 429 BUS STRUCTURE. THIS EQUIPMENT WAS DEVELOPED OVER A PERIOD OF 4 YEARS AND IS OPERATING VERY SUCCESSFULLY IN REVENUE SERVICE ON THE 757 AND 767 AIRPLANES. MANY PROBLEMS SURFACED DURING THIS DEVELOPMENT PERIOD, AND FIXES WERE FOUND, TESTED AND INCORPORATED. METHODOLOGY, AS WELL AS THE HARDWARE/SOFTWARE, WAS DEVELOPED, TESTED AND REFINED. STANDARDS WERE WRITTEN, IMPOSED AND REFINED. THIS INTENSIVE ACTIVITY PROVIDES AN INVALUABLE FUND OF EXPERIENCE ON WHICH FUTURE DIGITAL SYSTEM DEVELOPMENTS CAN BE BASED. THE MAJOR PROBLEMS ENCOUNTERED, THE LESSONS LEARNED IN RESOLVING THEM AND IN BRINGING THIS COMPLEX SYSTEM TO A SUCCESSFUL CONCLUSION ARE SUMMARIZED IN THIS PAPER.

LOCAL NETWORKS: FIBER OPTICS GAINS MOMENTUM
DOC. TYPE: ARTICLE
DOC. DATE: JUNE 23, 1983
AUTHORS: ALLAN, R.
SOURCE, PUB. BY, ETC: ELECTRONIC DESIGN, 06/23/83
FILED BY (NAME): KEMP, A.

LOCUS
DOC. TYPE: PAPER
DOC. DATE: DECEMBER 1, 1981
AUTHORS: POPEK, G. ET AL
SOURCE, PUB. BY, ETC: PROC 8TH SYMPOSIUM ON OPERATING SYSTEM PRINCIPLES
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
LOCUS IS A DISTRIBUTED OPERATING SYSTEM THAT PROVIDES A VERY HIGH DEGREE OF NETWORK TRANSPARENCY WHILE AT THE SAME TIME SUPPORTING HIGH PERFORMANCE AND AUTOMATIC REPLICATION OF STORAGE. BY NETWORK TRANSPARENCY WE MEAN THAT AT THE SYSTEM CALL INTERFACE THERE IS NO NEED TO MENTION ANYTHING NETWORK RELATED. KNOWLEDGE OF THE NETWORK AND CODE TO INTERACT WITH FOREIGN SITES IS BELOW THIS INTERFACEAND IS THUS HIDDEN FROM BOTH USERS AND PROGRAMS UNDER NORMAL CONDITIONS. LOCUS IS APPLICATION CODE COMPATIBLE WITH UNIX. AND SYSTEM UNIX. LOCUS RUNS ON A HIGH BANDWIDTH. LOW DELAY LOCAL NETWORK. IT IS DESIGNED TO PERMIT BOTH A SIGNIFICANT DEGREE OF LOCAL AUTONOMY FOR EACH SITE IN THE NETWORK WHILE STILL PROVIDING A NETWORK-WIDE, LOCATION INDEPENDENT NAME STRUCTURE. ATOMIC FILE OPERATIONS AND EXTENSIVE SYNCHRONIZATION ARE SUPPORTED.

SMALL, SLOW SITES WITHOUT LOCAL MASS STORE CAN COEXIST IN THE SAME NETWORK WITH MUCH LARGER AND MORE POWERFUL MACHINES WITHOUT LARGER MACHINES BEING SLOWED DOWN THROUGH FORCED INTERACTION WITH SLOWER ONES. GRACEFUL OPERATION DURING NETWORK TOPOLOGY CHANGES IS SUPPORTED.

MANAGEMENT GUIDE FOR INDEPENDENT VERIFICATION AND VALIDATION (IV&V)
DOC. TYPE: PAMPHLET
DOC. DATE: AUGUST 1980
AUTHORS: HENRY, R.
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THE PURPOSE OF THIS PAMPHLET IS TO PROVIDE INSIGHT INTO HOW THE NEED FOR AN INDEPENDENT VERIFICATION & VALIDATION (IV&V) CONTRACTOR IS ESTABLISHED, THE SCOPE OF THE IV&V EFFORT RELATIVE TO THE SIZE OF THE PROJECT ITSELF, HOW THE RFP SHOULD BE WRITTEN, WHAT CDRL ITEMS TO CALL FOR, WHAT TO LOOK FOR IN SOURCE SELECTIONS AND HOW TO MANAGE THE IV&V CONTRACT.

MARK1 - MARKOV MODELING PACKAGE
DOC. TYPE: REPORT
DOC. DATE: MARCH 1983
AUTHORS: LALA, JAYNARAYAN H.
SOURCE, PUB. BY, ETC: CSDL
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE PURPOSE OF THIS DOCUMENT IS TO ACQUAINT THE READER WITH THE FUNDAMENTAL MATHEMATICAL CONCEPTS UNDERLYING THE CHOSEN MODELING METHODOLOGY, TO DESCRIBE THE MODEL SPECIFICATION LANGUAGE FOR THE USE OF THE COMPUTER PROGRAMS AND TO ILLUSTRATE THE PROCEDURE FOR DOING THE COMBINATORIAL ANALYSIS OF THE SUBSYSTEM MODELS. THESE PROCEDURES ARE ILLUSTRATED WHERE NECESSARY WITH THE AID OF SIMPLE EXAMPLES. A STEP-WISE PROCEDURE HAS ALSO BEEN INCLUDED THAT SHOWS THE IMPORTANT STEPS IN GOING FROM A SYSTEM DESCRIPTION TO THE FINAL STAGE OF OBTAINING THE NUMERICAL RESULTS, USING A SIMPLE SYSTEM AS AN EXAMPLE.

MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR
DOC. TYPE: REPORT
DOC. DATE: OCTOBER 1981
AUTHORS: MCGOUGH, J. G. SWERN, F.
SOURCE, PUB. BY, ETC: BENDIX CORP.
DOCUMENT NUMBER: NASA CR - 3462
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS REPORT DESCRIBES THE RESULTS OF FAULT INJECTION EXPERIMENTS UTILIZING A GATE-LEVEL EMULATION OF THE CENTRAL PROCESSOR UNIT OF THE BENDIX BDX-930 DIGITAL COMPUTER. THE PRIMARY OBJECTIVE OF THE STUDY WAS TO ASCERTAIN THE FAILURE DETECTION COVERAGE OF COMPARISON-MONITORING AND A TYPICAL AVIONICS CPU SELF-TEST PROGRAM.

THE SPECIFIC TASKS AND EXPERIMENTS INCLUDED:

1. INJECT RANDOMLY SELECTED GATE-LEVEL AND PIN-LEVEL FAULTS AND EMULATE SIX SOFTWARE PROGRAMS USING COMPARISON-MONITORING TO DETECT THE FAULTS.

2. BASED UPON THE DERIVED EMPIRICAL DATA, DEVELOP AND VALIDATE A MODEL OF FAULT LATENCY THAT WILL FORECAST A SOFTWARE PROGRAM'S DETECTING ABILITY.

3. GIVEN A TYPICAL AVIONICS SELF-TEST PROGRAM, INJECT RANDOMLY SELECTED FAULTS AT BOTH THE GATE-LEVEL AND PIN-LEVEL AND DETERMINE THE PROPORTION OF FAULTS DETECTED.

4. DETERMINE WHY FAULTS WERE UNDETECTED.

5. RECOMMEND HOW THE EMULATION CAN BE EXTENDED TO MULTIPROCESSOR SYSTEMS SUCH AS SIFT.

6. DETERMINE THE PROPORTION OF FAULTS DETECTED BY A UNIPROCESSOR BIT (BUILT-IN-TEST) IRRESPECTIVE OF SELF-TEST.

MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR - PART II
DOC. TYPE: REPORT
DOC. DATE: 1983
AUTHORS: MCGOUGH, J. G. SWERN, F.
SOURCE, PUB. BY, ETC: BENDIX CORP.
DOCUMENT NUMBER: NASA CR-3651
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS REPORT DESCRIBES THE RESULTS OF FAULT INJECTION EXPERIMENTS UTILIZING A GATE LEVEL EMULATION OF THE CENTRAL PROCESSOR UNIT OF THE BENDIX BDX-930 DIGITAL COMPUTER. THE STUDY IS AN EXTENSION OF A PREVIOUS STUDY:

MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINI PROCESSOR NASA CR-3462, OCTOBER, 1981.

THE POOR COVERAGE OF COMPARISON-MONITORING, WHICH THE EARLIER STUDY DEMONSTRATED, COULD HAVE BEEN DUE TO THE LIMITED REPERTOIRE OF THE INSTRUCTION SET USED. AS A CONSEQUENCE, IT WAS DECIDED TO REPROGRAM SEVERAL EARLIER PROGRAMS BUT THIS TIME EXPANDING THE INSTRUCTION SET TO CAPITALIZE ON THE FULL POWER OF THE BDX-930 COMPUTER. AS A FINAL DEMONSTRATION OF

MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR - PART II * CONTINUED *

FAULT COVERAGE AN EXTENSIVE, 3-AXIS, HIGH PERFORMANCE FLIGHT CONTROL COMPUTATION WAS ADDED.

A SECONDARY OBJECTIVE OF THE STUDY WAS TO DEMONSTRATE THE STAGES IN THE DEVELOPMENT OF A CPU SELF-TEST PROGRAM EMPHASIZING THE RELATIONSHIP BETWEEN FAULT COVERAGE, SPEED AND QUANTITY OF INSTRUCTIONS.

MEASURES OF MERIT FOR FAULT-TOLERANT SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: GAI, E. G. ADAMS, M.
SOURCE, PUB. BY, ETC: CSDL
DOCUMENT NUMBER: CSDL P-1752
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
FAULT-TOLERANT SYSTEMS RELY ON REDUNDANCY TO ACHIEVE HIGH RELIABILITY. REDUNDANCY IMPLIES THAT THE SYSTEM CAN OPERATE IN MANY DIFFERENT MODES. THE VALUES OF A GIVEN PERFORMANCE MEASURE FOR THE SYSTEM OPERATING IN EACH OF THESE MODES ARE LIKELY TO BE DIFFERENT. IN THIS PAPER THE EVOLUTION OF THESE VALUES FOR ANY GIVEN MEASURE IS CHARACTERIZED AS A STOCHASTIC PROCESS, AND IT IS SUGGESTED THAT THE PERFORMANCE OF A FAULT-TOLERANT SYSTEM BE EVALUATED AND SPECIFIED IN TERMS OF A SCALAR FUNCTION OF THE PROBABILITY LAW OF THAT STOCHASTIC PROCESS.

METHODOLOGY FOR MEASUREMENT OF FAULT LATENCY IN A DIGITAL AVIONIC MINIPROCESSOR

DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: MCGOUGH, J. G. SWERN, F. BAVUSO, S. J.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE, 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
USING A GATE-LEVEL EMULATION OF A TYPICAL AVIONICS MINIPROCESSOR, FAULT INJECTION EXPERIMENTS WERE PERFORMED TO (1) DETERMINE THE TIME-TO-DETECT A FAULT BY COMPARISON-MONITORING, (2) FORECAST A PROG'S ABILITY TO DETECT FAULTS AND (3) VALIDATE THE FAULT DETECTION COVERAGE OF A TYPICAL SELF-TEST PROG.

TO ESTIMATE TIME-TO-DETECT, 6 PROGS RANGING IN COMPLEXITY FROM 6 TO 147 INSTRUCTIONS, WERE EMULATED. EACH PROG WAS EXECUTED REPETITIVELY IN THE PRESENCE OF A SINGLE STUCK-AT FAULT AT A GATE NODE OR DEVICE PIN. DETECTION WAS ASSUMED TO OCCUR WHENEVER THE COMPUTED OUTPUTS DIFFERED FROM THE CORRESPONDING OUTPUTS OF THE SAME PROG EXECUTED IN A NON-FAULTED PROCESSOR. HISTOGRAMS OF FAULTS DETECTED VERSUS NUMBER OF REPETITIONS TO DETECTION WERE TABULATED.

USING A SIMPLE MODEL OF FAULT DETECTION, WHICH WAS BASED ON AN ANALOGY WITH THE SELECTION OF BALLS IN AN URN, DISTRIBUTIONS OF TIME-TO-DETECT WERE COMPUTED AND COMPARED WITH THOSE OBTAINED EMPIRICALLY.

A SELF-TEST PROGRAM OF 2,000 EXECUTABLE INSTRUCTIONS WAS DESIGNED EXPRESSLY FOR THE STUDY. THE ONLY REQUIREMENT IMPOSED ON THE DESIGN WAS THAT IT SHOULD ACHIEVE 95% COVERAGE. THE PROPORTION OF DETECTED FAULTS WAS TABULATED.

MICRO-ANALYSIS OF COMPUTER SYSTEM PERFORMANCE

DOC. TYPE: BOOK
DOC. DATE: 1978
AUTHORS: BEIZER, B.
SOURCE, PUB. BY, ETC: VAN NOSTRAN REINHOLD, WOKINGHAM, ENGLAND
DOCUMENT NUMBER: ISBN 0 442 20663 1
FILED BY (NAME): LALA, J. H.

MICROPROCESSOR SYSTEMS AND ARCHITECTURES FOR APPLICATIONS TO THE CONTROL AND PROTECTION OF ELECTRIC POWER SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: 1979
AUTHORS: RUMMER, D. I. KEZUNOVIC, M.
SOURCE, PUB. BY, ETC: ACM
DOCUMENT NUMBER: 0-89791-007-9/79/1000-0001
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
SOME DESIGN CONCEPTS FOR MICROPROCESSOR APPLICATIONS TO THE CONTROL AND PROTECTION OF ELECTRIC POWER SYSTEMS ARE PRESENTED. A FORMAL DESIGN METHODOLOGY IS OUTLINED. PRESENT MICROPROCESSOR APPLICATIONS TO THE CONTROL AND PROTECTION OF THE ELECTRIC POWER GENERATION, TRANSMISSION AND DISTRIBUTION SYSTEMS ARE SUMMARIZED. THESE APPLICATIONS ARE GENERALIZED TO PROVIDE A BASIS FOR IDENTIFYING A SET OF "STANDARDIZED APPLICATION MODULES". A PROJECTION IS MADE OF PROBABLE FUTURE DEVELOPMENTS IN MICROPROCESSOR APPLICATIONS TO THE ELECTRIC POWER INDUSTRY. A PROBABLE DEVELOPMENT IS BELIEVED TO BE THE

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

MICROPROCESSOR SYSTEMS AND ARCHITECTURES FOR APPLICATIONS TO THE CONTROL AND * CONTINUED *
USE OF HIERARCHICALLY STRUCTURED NETWORKS OF MICROPROCESSORS TO PROVIDE, THROUGH DISTRIBUTED PROCESSING, MORE COST-EFFECTIVE AND RELIABLE CONTROL AND PROTECTION THAN IS POSSIBLE THROUGH CENTRALIZED PROCESSING.

MIL-STD-1553 MULTIPLEX DATA BUS WORD FORMATS - FINAL REPORT FOR PERIOD NOVEMBER 1980 - DECEMBER 1981
DOC. TYPE: FINAL REPORT
DOC. DATE: DECEMBER 1981
SOURCE, PUB. BY, ETC: BOEING MIL. AIRPLANE CO., SEATTLE, WA 98124
DOCUMENT NUMBER: TIC# ADA121934
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
THE EMPHASIS IN THIS REPORT IS THE DEVELOPMENT OF DATA WORD AND MESSAGE FORMATS FOR MIL-STD-1553 DATA BUS APPLICATIONS. THIS REPORT IS INTENDED AS A GUIDE TO THE DESIGNER TO IDENTIFY STANDARD DATA WORDS AND MESSAGES THAT ARE BEING USED IN TODAY'S AVIONIC SYSTEMS AND SUBSYSTEMS. THESE STANDARD WORDS AND MESSAGES, AS WELL AS THE DOCUMENTATION FORMAT FOR INTERFACE CONTROL DOCUMENT (ICD) SHEETS, PROVIDE THE BASIS FOR DEFINING NEW 1553 SYSTEMS. THE STANDARDS DEFINED IN THIS REPORT HAVE MET THE TEST OF APPLICATION USAGE ACROSS SEVERAL 1553 SYSTEMS AND THUS PROVIDE THE ACCEPTABLE METHOD FOR TRANSMITTING SIGNALS OF THIS TYPE IN A 1553 SYSTEM. ALSO PROVIDED IN THIS REPORT IS THE METHOD FOR DEVELOPING ADDITIONAL DATA WORD FORMATS AND MESSAGES THAT MAY BE REQUIRED BY A PARTICULAR SYSTEM BUT ARE NOT COVERED BY THE STANDARDS PROVIDED HEREIN. IT IS ESSENTIAL THAT ANY NEW WORD FORMATS OR MESSAGE FORMATS THAT ARE DEVELOPED FOR A 1553 APPLICATION FOLLOW THE FUNDAMENTAL GUIDELINES ESATBLISHED IN THIS REPORT IN ORDER TO EASE FUTURE STANDARDIZATION OF THESE WORDS AND MESSAGES IF EXPERIENCE AND USAGE DEMAND IT.

MIL-STD-1750A CPU
DOC. TYPE: DATA SHEET
DOC. DATE: NOVEMBER 30, 1982
SOURCE, PUB. BY, ETC: MCDONNEL-DOUGLAS ASTRONAUTICS
FILED BY (NAME): KEMP, A.

MIL-STD-1750A MICROPROCESSOR
DOC. TYPE: DATA SHEET
DOC. DATE: NOVEMBER 30, 1982
SOURCE, PUB. BY, ETC: FAIRCHILD
FILED BY (NAME): KEMP, A.

MIL-STD-1815A (ADA) AND MIL-STD-1750A: PROBLEMS AND SOLUTIONS
DOC. TYPE: PAPER
DOC. DATE: AUGUST 1983
AUTHORS: EHRENFRIED, LT. DANIEL
SOURCE, PUB. BY, ETC: AFWAL/AAAF-2 WPAFB, OH 45433 (513) 255-2446
FILED BY (NAME): KNOSP, A. A.
ABSTRACT:
TWO FACTS ARE QUICKLY BECOMING APPARENT WITHIN THE AIR FORCE'S MOVE TOWARD STANDARDIZATION. MIL-STD-1750A, EVEN WITH ITS SHORTCOMINGS, HAS REMIANED THE INSTRUCTION SET ARCHITECTURE STANDARD OF CHOICE FOR EMBEDDED SYSTEM SPECIFICATIONS. INDUSTRY HAS ALSO, FOR THE MOST PART, PLACED ITS SUPPORT BEHIND THIS STANDARD AND 1750A HARDWARE IS MOVING INTO OPERATIONAL INVENTORIES. IT IS THEREFORE REASONABLE TO EXPECT THAT EMBEDDED SYSTEMS WILL INCLUDE THE USE OF 1750A FOR SEVERAL YEARS TO COME. THE ADA PROGRAMMING LANGAUGE STANDARD IS EMERGING WITH A BROAD BASE OF SUPPORT FROM BOTH INDUSTRY AND ACADEMIA AS WELL AS ITS TRI-SERVICE SPONSORS. ADA'S MOMENTUM SHOULD MAKE IT THE STANDARD OF CHOICE TO REPLACE JOVIAL IN THE NEAR FUTURE. THIS PAPER WILL ATTEMP TO ATICIPATE SOME OF THE DIFFICULTIES IN MERGING THESE TWO STANDARDS INTO A WORKING SYSTEM FOR USE IN EMBEDDED APPLICATIONS. ALTHOUGH SEVERAL OF THE CITED PROBLEMS ARE INDEED SERIOUS, AN ACCEPTABLE SOLUTION OR WORK AROUND IS OFFERED IN EACH INSTANCE.

MODELING AND PERFORMANCE EVALUATION OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, ALGIRDAS
SOURCE, PUB. BY, ETC: ??
FILED BY (NAME): FURTEK, F. C.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

MODELING OF SOFTWARE FAULT-TOLERANT STRATEGIES
DOC. TYPE: PAPER
DOC. DATE: MAY 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.
SOURCE, PUB. BY, ETC: 11TH ANN PTSBG CONF ON MDLG SIMLT & PROCESSING
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
THE PAPER PRESENTS MODELS FOR THE EVALUATION OF SOFTWARE FAULT-TOLERANCE STRATEGIES. IN PARTICULAR, MODELS FOR EVALUATING PROCESSING TIME AND RELIABILITY PERFORMANCES OF THE TWO MAIN STRATEGIES, THE RECOVERY BLOCK SCHEME AND N-VERSION PROGRAMMING, ARE DERIVED AND ANALYZED. THE OBTAINED RESULTS CAN BE USEFUL FOR SYSTEM DESIGNERS IN DECIDING WHICH OF THE STRATEGIES SHOULD BE USED, DEPENDING ON SYSTEM PARAMETERS.

MODELS AND MEASUREMENTS OF PARALLELISM FOR A DISTRIBUTED COMPUTER SYSTEM
DOC. TYPE: THESIS
AUTHORS: LANE, D.
DOCUMENT NUMBER: TIC# AD-A116762
FILED BY (NAME): FELLEMAN, P. G.
ABSTRACT:
DISTRIBUTED COMPUTER SYSTEMS MAY BE DEFINED AS LOCAL COMPUTER NETWORKS EXECUTING DECENTRALIZED SOFTWARE. THE DECENTRALIZATION OF SOFTWARE IS ACCOMPLISHED BY LOGICALLY DISTRIBUTING THE CONTROL AND INFORMATION STRUCTURES AMONG THE COMPUTERS. THE PROGRAMMING LANGUAGE CONCEPTS NECESSARY TO DECENTRALIZE SOFTWARE IS CURRENTLY A TOPIC OF RESEARCH. ONE APPROACH TO CONSTRUCTING SUCH SOFTWARE IS THROUGH THE USE OF AUTONOMOUS PROCESSES. SYNCHRONIZATION AND COMMUNICATION BETWEEN PROCESSES IS ACHIEVED BY THE TRANSMISSION OF UNBUFFERED MESSAGES. A PROCESS MODEL, ALONG WITH THE ABILITY TO SEND AND RECEIVE MESSAGES IN A NON-DETERMINISTIC MANNER ARE THE MOST IMPORTANT PROGRAMMING CONCEPTS NECESSARY TO DISTRIBUTE SOFTWARE.

MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS
DOC. TYPE: REPORT
DOC. DATE: JULY 1982
AUTHORS: MEYER, J. F.
SOURCE, PUB. BY, ETC: THE UNIVERSITY OF MICHIGAN, SYSTEMS ENGR'G LAB
DOCUMENT NUMBER: SYSTEMS ENGR LAB REPORT SEL170
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE PURPOSE OF THIS RESEARCH PROJECT WAS TO DEVELOP MODELS, MEASURES, AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS. BY "EFFECTIVENESS" IN THIS CONTEXT WE MEAN THE EXTENT TO WHICH THE USER, I.E., A COMMERCIAL AIR CARRIER, MAY EXPECT TO BENEFIT FROM THE COMPUTATIONAL TASKS ACCOMPLISHED BY A COMPUTING SYSTEM IN THE ENVIRONMENT OF AN ADVANCED COMMERCIAL AIRCRAFT. THUS, THE CONCEPT OF EFFECTIVENESS INVOLVES ASPECTS OF SYSTEM PERFORMANCE, RELIABILITY, AND WORTH (VALUE, BENEFIT) WHICH MUST BE APPROPRIATELY INTEGRATED IN THE PROCESS OF EVALUATING SYSTEM EFFECTIVENESS. SPECIFICALLY, THE PRIMARY OBJECTIVES OF THIS PROJECT ARE:

I. THE DEVELOPMENT OF SYSTEM MODELS THAT CAN PROVIDE A BASIS FOR THE FORMULATION AND EVALUATION OF AIRCRAFT COMPUTER SYSTEM EFFECTIVENESS.

II. THE FORMULATION OF QUANTITATIVE MEASURES OF SYSTEM EFFECTIVENESS, AND

III. THE DEVELOPMENT OF ANALYTIC AND SIMULATION TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF A PROPOSED OR EXISTING AIRCRAFT COMPUTER.

MODELS FOR EVALUATING THE PERFORMABILITY OF DEGRADABLE COMPUTING SYSTEMS
DOC. TYPE: REPORT
DOC. DATE: JUNE 1982
AUTHORS: WU, L.
SOURCE, PUB. BY, ETC: UNIV. OF MICHIGAN, COMPUTING RESEARCH LAB.
DOCUMENT NUMBER: CRL-TR-7-82
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
RECENT ADVANCES IN MULTIPROCESSOR TECHNOLOGY HAVE ESTABLISHED THE NEED FOR UNIFIED METHODS TO EVALUATE COMPUTING SYSTEMS PERFORMANCE AND RELIABILITY. IN RESPONSE TO THIS MODELING NEED, THIS DISSERTATION CONSIDERS A GENERAL MODELING FRAMEWORK THAT PERMITS THE MODELING, ANALYSIS AND EVALUATION OF DEGRADABLE COMPUTING SYSTEMS. WITHIN THIS FRAMEWORK, SEVERAL USER-ORIENTED PERFORMANCE VARIABLES ARE IDENTIFIED AND SHOWN TO BE PROPER GENERALIZATIONS OF THE TRADITIONAL NOTIONS OF SYSTEM PERFORMANCE AND RELIABILITY. FUTHERMORE, A TIME-VARYING VERSION OF THE MODEL IS DEVELOPED TO GENERALIZE THE TRADITIONAL FAULT-TREE RELIABILITY EVALUATION METHODS OF PHASED MISSIONS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

MULTIPLE MICROPROCESSOR SYSTEMS: WHAT, WHY, AND WHEN
DOC. TYPE: ARTICLE
DOC. DATE: MARCH 1983
AUTHORS: FATHI, E. KRIEGER, M.
SOURCE, PUB. BY, ETC: UNIVERSITY OF OTTAWA
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
MULTIPLE MICROPROCESSOR SYSTMS CAN PROVIDE AN APPROPRIATE SOLUTION TO THE DEMAND FOR ADDITIONAL COMPUTING POWER TO MEET NEW REQUIREMENTS AND TO SUPPORT COMPLEX APPLICATIONS. TO CLARIFY THE CONCEPT AND ITS ASSOCIATED TERMINOLOGY, THIS ARTICLE CONSIDERS THE "WHAT," "WHY," AND "WHEN" OF MULTIPLE MICROPROCESSOR SYSTMS. ASPECTS THAT APPLY TO ALL PROCESSORS, REGARDLESS OF SIZE, ARE PRESENTED IN GENERAL TERMS. HOWEVER, SINCE OUR MAIN INTEREST LIES WITH MICROPROCESSOR-BASED SYSTMS, ASPECTS THAT DEPEND ON PROCESSOR POWER AND I/O FLEXIBILITY ARE RELATED SPECIFICALLY TO MICROPROCESSORS.

AS THE NUMBER OF APPLICATIONS WITH MORE ELABORATE COMPUTATIONAL DEMANDS INCREASES, WE NEED TO PROVIDE MORE PROCESSING POWER. THIS CAN BE ACHIEVED AT THE PROCESSOR LEVEL, BY RELYING ON TECHNOLOGICAL IMPROVEMENTS TO PUSH THE MICROPROCESSOR BEYOND ITS CURRENT MAX CAPABILITIES, OR AT THE SYS LEVEL, BY EXTENDING THE CAPABILITIES OF A SINGLE MICROPROCESSOR THROUGH CONCURRENT EXECUTION.

THERE ARE DEFINITE LIMITS TO THE EXTENT AND TYPE OF IMPROVEMENTS POSSIBLE WITH EXISTING TECHNOLOGY. THUS, UNLESS NEW TECHNOLOGY IS DEVELOPED, ALL TECHNOLOGICAL IMPROVEMENTS CAN BE REGARDED AS EVOLUTIONARY RATHER THAN REVOLUTIONARY

MULTIPROCESSORS AND PARALLEL PROCESSING
DOC. TYPE: BOOK
DOC. DATE: 1974
AUTHORS: ENSLOW, PHILIP H., JR.
SOURCE, PUB. BY, ETC: JOHN WILEY & SONS, INC. (PREP: THE COMTRE CORP.)
FILED BY (NAME): LALA, J. H.
ABSTRACT:
(EXCERPT FROM THE PREFACE BY P. ENSLOW, EDITOR:)

THIS TEXT HAS SEVERAL UNUSUAL UNIQUE FEATURES.

1) IT IS THE ONLY WORK DEVOTED ENTIRELY TO A DISCUSSION OF MULTIPROCESSORS.

2) IT IS THE ONLY SOURCE PROVIDING A COMPLETE TREATMENT OF MULTIPROCESSOR SYSTEM ORGANIZATION COVERING BOTH HARDWARE AND SOFTWARE.

3) IT ILLUSTRATES HOW THE MULTIPROCESSOR IS A LOGICAL RESULT OF THE EFFORTS TO INCREASE COMPUTER PERFORMANCE BY THE USE AND EXPLOITATION OF CONCURRENCY AND PARALLELISM IN BOTH PROGRAM EXECUTION AND IN HARDWARE.

4) IT PROVIDES A SUCCINCT DISCUSSION OF MOST OF THE MAJOR MULTIPROCESSOR SYSTEMS NOW IN PRODUCTION.

N-VERSION PROGRAMMING: A FAULT-TOLERANCE APPROACH TO RELIABILITY OF SOFTWARE OPERATION
DOC. TYPE: PAPER
DOC. DATE: JUNE 21, 1978
AUTHORS: CHEN, L. AVIZIENIS, A.
SOURCE, PUB. BY, ETC: 8TH ANN INTL CONF ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
N-VERSION PROGRAMMING IS DEFINED AS THE INDEPENDENT GENERATION OF N > 2 FUNCTIONALLY EQUIVALENT PROGRAMS FROM THE SAME INITIAL SPECIFICATION. A METHODOLOGY OF N-VERSION PROGRAMMING HAS BEEN DEVISED AND THREE TYPES OF SPECIAL MECHANISMS HAVE BEEN IDENTIFIED THAT ARE NEEDED TO COORDINATE THE EXECUTIONOF AN N-VERSION SOFTWARE UNIT AND TO COMPARE THE CORRESPONDENT RESULTS GENERATED BY EACH VERSION. TWO EXPERIMENTS HAVE BEEN CONDUCTED TO TEST THE FEASIBILITY OF N-VERSION PROGRAMMING. THE RESULTS OF THESE EXPERIMENTS ARE DISCUSSED. IN ADDITION, CONSTRAINTS ARE IDENTIFIED THAT MUST BE MET FOR EFFECTIVE APPLICATION OF N-VERSION PROGRAMMING.

NEXT GENERATION MILITARY AIRCRAFT WILL REQUIRE HIERARCHICAL/MULTILEVEL INFORMATION TRANSFER SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: MCCUEN, J. W.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
TIC #AD-A109-274. THE AUTHOR IS WITH HUGHES AIRCRAFT COMPANY, FULLERTON, CALIFORNIA USA.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

OBJECT-ORIENTED DESCRIPTION ENVIRONMENT FOR COMPUTER HARDWARE
DOC. TYPE: PAPER
AUTHORS: TAKEUCHI, A.
SOURCE, PUB. BY, ETC: IFIP 5TH INTN'L CONF, 7-9 SEPT 1981
DOCUMENT NUMBER: ISBN 0 444 86279 X
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE SYSTEM NAMED OODE IS PRESENTED. OODE IS BASED ON THE SO-CALLED OBJECT ORIENTED LANGUAGES AND A HIGH DEGREE OF MODULE STRUCTURE IS ACHIEVED WITH IT. EVERY MODULE IS DESCRIBED FROM THREE POINTS OF VIEW, BEHAVIORAL, STRUCTURAL AND CONCEPTUAL. A MODULE IS DEFINED BY REFERENCING TO AN ABSTRACT MODULE WITH ITS PARAMETERS SPECIFIED (CONCEPTUAL DESCRIPTION) OR BY CONSTRUCTING FROM OTHER MODULES (STRUCTURAL DESCRIPTION). BEHAVIORAL DESCRIPTION OF A MODULE IS BASED ON THE MESSAGE PASSING.

THE PAPER IS FROM COMPUTER HARDWARE DESCRIPTION LANGUAGES AND THEIR APPLICATIONS, M. BREUR & R. HARTENSTEIN, (EDS), PUB BY N.-HOLLAND PUB'G CO., AMSTERDAM, THE NETHERLANDS.

THE AUTHOR IS WITH THE CENTRAL RESEARCH LABORATORY, MITSUBISHI ELECTRIC CORPORATION, AMAGASAKI, HYOGO, JAPAN.

OBJECT-ORIENTED SOFTWARE SYSTEMS
DOC. TYPE: ARTICLE
DOC. DATE: AUGUST 1981
AUTHORS: ROBSON, D.
SOURCE, PUB. BY, ETC: BYTE, V 6, #8
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS ARTICLE DESCRIBES A GENERAL CLASS OF TOOLS FOR MANIPULATING INFORMATION CALLED OBJECT-ORIENTED SOFTWARE SYSTEMS. IT DEFINES A SERIES OF TERMS, INCLUDING SOFTWARE SYSTEM AND OBJECT-ORIENTED. THE DESCRIPTION IS GREATLY INFLUENCED BY A SERIES OF OBJECT-ORIENTED PROGRAMMING ENVIRONMENTS DEVELOPED IN THE LAST TEN YEARS BY THE LEARNING RESEARCH GROUP OF XEROX'S PALO ALTO RESEARCH CENTER, THE LATEST BEING THE SMALLTALK-80 SYSTEM. THE ARTICLE DESCRIBES OBJECT-ORIENTED SOFTWARE SYSTEMS IN GENERAL, INSTEAD OF THE SMALLTALK-80 SYSTEM IN PARTICULAR, IN ORDER TO FOCUS ATTENTION ON THE FUNDAMENTAL PROPERTY THAT SETS THE SMALLTALK-80 SYSTEM APART FROM MOST OTHER PROGRAMMING ENVIRONMENTS. THE WORDS "OBJECT-ORIENTED" MEAN DIFFERENT THINGS TO DIFFERENT PEOPLE. ALTHOUGH THE DEFINITION GIVEN IN THIS ARTICLE MAY EXCLUDE SYSTEMS THAT SHOULD RIGHTFULLY BE CALLED OBJECT-ORIENTED, IT IS A USEFUL ABSTRACTION OF THE IDEA BEHIND MANY SOFTWARE SYSTEMS.

OBJECT-ORIENTED STRUCTURED DESIGN OF LAYERED PROTOCOL SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1982
AUTHORS: BUHR, R. J. A. MICHELL, S.
SOURCE, PUB. BY, ETC: IEEE, 3RD INTNL CONF ON DISTRBTD CMPTGSYSTS, 1982
DOCUMENT NUMBER: IEEE CH1802-8/82/0000/0288
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER ILLUSTRATES WITH A CASE STUDY HOW OBJECT ORIENTED STRUCTURED DESIGN PROVIDES AN ELEGANT AND PRACTICAL APPROACH TO DESIGNING AND DESCRIBING LAYERED PROTOCOL SYSTEMS. THE APPROACH IS A COMBINATION AND EXTENSION OF YOURDON'S STRUCTURED DESIGN METHODOLOGY AND THE GRAPHICAL SYSTEM DESCRIPTION TECHNIQUES USED BY INTEL CORPORATION AND OTHERS TO DESCRIBE OBJECT INTERACTIONS IN IAPX432 HARDWARE AND ADA SOFTWARE. THE APPROACH IS FIRST INTRODUCED BY A SIMPLE PHYSICAL EXAMPLE AND THEN APPLIED TO DESCRIBE SEVERAL DIFFERENT ORGANIZATIONS FOR A REALISTIC LAYERED PROTOCOL EXAMPLE, NAMELY X.25.

ON EVALUATING THE PERFORMABILITY OF DEGRADABLE COMPUTING SYSTEMS
DOC. TYPE: ARTICLE
DOC. DATE: AUGUST 1980
AUTHORS: MEYER, J. F.
SOURCE, PUB. BY, ETC: IEEE TRANS. ON COMPUTERS, VOL C-29, NO.8
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IF THE PERFORMANCE OF A COMPUTING SYSTEM IS "DEGRADABLE," PERFORMANCE AND RELIABILITY ISSUES MUST BE DEALT WITH SIMULTANEOUSLY IN THE PROCESS OF EVALUATING SYSTEM EFFECTIVENESS. FOR THIS PURPOSE, A UNIFIED MEASURE, CALLED "PERFORMABILITY," IS INTRODUCED AND THE FOUNDATIONS OF PERFORMABILITY MODELING AND EVALUATION ARE ESTABLISHED. A CRITICAL STEP IN THE MODELING PROCESS IS THE INTRODUCTION OF A "CAPABILITY FUNCTION" WHICH RELATES LOW-LEVEL SYSTEM BEHAVIOR TO USER-ORIENTED PERFORMANCE LEVELS. A HIERARCHICAL MODELING SCHEME IS USED TO FORMULATE THE CAPABILITY FUNCTION AND CAPABILITY IS USED, IN TURN, TO EVALUATE PERFORMABILITY. THESE TECHNIQUES ARE THEN ILLUSTRATED FOR A SPECIFIC APPLICATION: THE PERFORMABILITY EVALUATION OF AN AIRCRAFT COMPUTER IN THE ENVIRONMENT OF AN AIR TRANSPORT MISSION.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

ON THE IMPLEMENTATION OF N-VERSION PROGRAMMING FOR SOFTWARE FAULT-TOLERANCEDURING PROGRAM EXECUTION
DOC. TYPE: ARTICLE
DOC. DATE: NOVEMBER 8, 1977
AUTHORS: AVIZIENIS, A. CHEN, L.
SOURCE, PUB. BY, ETC: IEEE COMP SOC 1ST INT COMP SFTWR & APPLCTNS CONF
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
N-VERSION PROGRAMMING RESULTS IN N INDEPENDENTLY GENERATED, BUT FUNCTIONALLY EQUIVALENT PROGRAMS WHICH ARE INTENDED TO PROVIDE FAULT-TOLERANCE FOR SOFTWARE FAULTS DURING PROGRAM EXECUTION. A PILOT EXPERIMENT IN N-VERSION PROGRAMMING IS DESCRIBED AND AN EVOLVING METHODOLOGY FOR THIS FORM OF PROGRAMMING IS OUTLINED.

THIS PAPER WAS PRESENTED AT THE IEEE COMPUTER SOCIETY FIRST INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC), IN CHICAGO, 8-11 NOVEMBER 1977. (PROCEEDINGS PP 149-155.)

ON THE PERFORMANCE OF SOFTWARE FAULT-TOLERANCE STRATEGIES
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1, 1980
AUTHORS: GRNAROV, A. ARLAT, J. AVIZIENIS, A.
SOURCE, PUB. BY, ETC: 10TH ANN INTL SYMP ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
IN THE PAPER A COMPARISON OF PROCESSING TIME AND RELIABILITY PERFORMANCE FOR THE RECOVERY BLOCKS SCHEME AND N-VERSION PROGRAMMING TECHNIQUE IS PRESENTED. DERIVED QUEUEING MODELS CAN BE USEFUL IN DECIDING WHICH OF THE STRATEGIES SHOULD BE USED, DEPENDING ON SYSTEM PARAMETERS.

OPTIMAL DESIGN OF MULTILEVEL STORAGE HIERARCHIES
DOC. TYPE: ARTICLE
DOC. DATE: MARCH 1982
AUTHORS: GEIST, R. TRIVEDI, K.
SOURCE, PUB. BY, ETC: IEEE TRANS ON COMPUTERS, VOL C-31, NO.3
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
AN OPTIMIZATION MODEL IS DEVELOPED FOR ASSIGNING A FIXED SET OF FILES ACROSS AN ASSEMBLAGE OF STORAGE DEVICES SO AS TO MAXIMIZE SYSTEM THROUGHPUT. MULTIPLE LEVELS OF EXECUTABLE MEMORIES AND DISTINCT RECORD SIZES FOR SEPARATE FILES ARE ALLOWED. THROUGH THE USE OF THIS MODEL, A GENERAL CLASS OF FILE ASSIGNMENT PROBLEMS IS REDUCED TO THE OPTIMIZATION OF A CONVEX FUNCTION OVER A CONVEX FEASIBLE REGION. A HIGH-SPEED SEARCH PROCEDURE SPECIFICALLY TAILORED TO SOLVE THIS OPTIMIZATION PROBLEM IS THEN PRESENTED, ALONG WITH NUMERICAL EXAMPLES FROM REAL SYSTEMS WHICH DEMONSTRATE ORDERS OF MAGNITUDE IMPROVEMENT IN EXECUTION TIME OVER EXISTING ROUTINES FOR SOLVING THE FILE-ASSIGNMENT PROBLEM. THE OPTIMAL DEVICE CAPACITY SELECTION PROBLEM IS THEN SOLVED BY SIMPLY CALLING THE FILE ASSIGNMENT ROUTINE FOR EACH CANDIDATE SET OF DEVICE CAPACITIES.

PDL - A TOOL FOR SOFTWARE DESIGN
DOC. TYPE: PAPER
DOC. DATE: 1975
AUTHORS: CAINE, S. H. GORDON, E. K.
SOURCE, PUB. BY, ETC: PROC. AFIPS 1975 NAT'L COMPTR CONF, PP 271-276
FILED BY (NAME): SZULEWSKI, P.

PERFORMABILITY EVALUATION OF COMPUTING SYSTEMS USING REWARD MODELS
DOC. TYPE: REPORT
DOC. DATE: AUGUST 1983
AUTHORS: FURCHTGOTT, D. MEYER, J. F.
SOURCE, PUB. BY,. ETC: COMPUTING RESEARCH LAB., UNIV. OF MICHIGAN
DOCUMENT NUMBER: CRL-TR-27-83
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
AN EVALUATION ALGORITHM IS DEVELOPED FOR A BROAD CLASS OF PERFORMABILITY MODELS WHEREIN SYSTEM PERFORMANCE IS IDENTIFIED WITH "REWARD." MORE PRECISELY, FOR A SYSTEM S AND A UTILIZATION PERIOD T, THE PERFORMANCE VARIABLE OF THE MODEL IS THE REWARD DERIVED FROM USING S DURING T. THE STATE BEHAVIOR OF S IS REPRESENTED BY A FINITE-STATE STOCHASTIC PROCESS (THE BASE MODEL); REWARD IS DETERMINED BY REWARD RATES ASSOCIATED WITH THE STATES OF THE BASE MODEL. IT IS ASSUMED THAT THE CORRESPONDING REWARD MODEL IS A NONRECOVERABLE PROCESS IN THE SENSE THAT A FUTURE STATE (REWARD RATE) OF THE MODEL CANNOT BE GREATER THAN THE PRESENT STATE. FOR THIS MODEL CLASS, WE OBTAIN A GENERAL METHOD FOR DETERMINING THE PROBABILITY DISTRIBUTION FUNCTION OF THE PERFORMANCE (REWARD) VARIABLE AND, HENCE, THE PERFORMABILITY OF THE CORRESPONDING SYSTEM. MOREOVER, THIS IS DONE FOR BOUNDED UTILIZATION PERIODS, AN ASSUMPTION WHICH DEMANDS A RELATIVELY COMPLEX SOLUTION. THE RESULT IS AN INTEGRAL EXPRESSION WHICH CAN BE SOLVED EITHER ANALYTICALLY OR NUMERICALLY. A PROGRAM WRITTEN FOR NUMERICAL SOLUTIONS IS DISCUSSED AND AN EXAMPLE ILLUSTRATING THE METHOD IS PRESENTED.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

PERFORMABILITY EVALUATION OF THE SIFT COMPUTER
DOC. TYPE: ARTICLE
DOC. DATE: JUNE 1980
AUTHORS: MEYER, J. F. FURCHTGOTT, D. WU, L.
SOURCE, PUB. BY, ETC: IEEE TRANS. ON COMPUTERS VOL C-29, NO.6
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
PERFORMABILITY MODELING AND EVALUATION METHODS ARE APPLIED TO THE SIFT COMPUTER IN THE COMPUTATIONAL ENVIRONMENT OF AN AIR TRANSPORT MISSION. USER-VISIBLE PERFORMANCE OF THE "TOTAL SYSTEM" (SIFT PLUS ITS ENVIRONMENT) IS MODELED AS A RANDOM VARIABLE TAKING VALUES IN A SET OF "ACCOMPLISHMENT LEVELS." THESE LEVELS ARE DEFINED IN TERMS OF FOUR ATTRIBUTES OF TOTAL SYSTEM BEHAVIOR: SAFETY, NO CHANGE IN MISSION PROFILE, NO OPERATIONAL PENALITIES, AND NO ECONOMIC PENALTIES. THE "BASE MODEL" OF THE TOTAL SYSTEM IS A STOCHASTIC PROCESS WHOSE STATES DESCRIBE THE INTERNAL STRUCTURE OF SIFT AS WELL AS RELEVANT CONDITIONS OF ITS ENVIRONMENT. BASE MODEL STATE TRAJECTORIES ARE RELATED TO ACCOMPLISHMENT LEVELS VIA A "CAPABILITY FUNCTION" WHICH IS FORMULATED IN TERMS OF A THREE-LEVEL MODEL HIERARCHY. SOLUTION METHODS ARE THEN APPLIED TO DETERMINE THE PERFORMABILITY OF THE TOTAL SYSTEM FOR VARIOUS CHOICES OF COMPUTER AND ENVIRONMENT PARAMETER VALUES.

PERFORMABILITY MODELING OF DISTRIBUTED REAL-TIME SYSTEMS
DOC. TYPE: REPORT
DOC. DATE: AUGUST 1983
AUTHORS: MEYER, J. F.
SOURCE, PUB. BY, ETC: COMPUTING RESEARCH LAB., UNIV. OF MICHIGAN
DOCUMENT NUMBER: CRL-TR-28-63
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS SURVEY/POSITION PAPER CONCERNS MODELING CONCEPTS AND TECHNIQUES FOR THE PERFORMABILITY (PERFORMANCE-RELIABILITY) EVALUATION OF DISTRIBUTED REAL-TIME SYSTEMS. DUE TO THE NATURE OF APPLICATION REQUIREMENTS, SUCH SYSTEMS TYPICALLY EXHIBIT PROPERTIES OF CONCURRENCY, TIMELINESS, FAULT TOLERANCE, AND DEGRADABLE PERFORMANCE. RELEVANT PRIOR RESEARCH IS SURVEYED AND CLASSIFIED ACCORDING TO THESE PROPERTIES AND, BASED ON THE POSITION THAT PERFORMABILITY MODELS SHOULD ACCOMMODATE ALL FOUR PROPERTIES, DIRECTIONS FOR FUTURE RESEARCH ARE SUGGESTED.

PERFORMANCE EVALUATION OF A FAULT-TOLERANT COMPUTING SYSTEM
DOC. TYPE: PAPER
DOC. DATE: JUNE 1979
AUTHORS: MINE, H. HATAYAMA, K.
SOURCE, PUB. BY, ETC: 9TH ANN INT SYM ON FAULT-TOLERANT COMPUTING
DOCUMENT NUMBER: CH1396-1/79/000-0059
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
WE CONSIDER A COMPUTING SYSTEM CONSISTING OF SUBSYSTEMS CONTAINING MANY IDENTICAL COMPONENTS AND EXECUTING VARIOUS TYPES OF JOBS. IT IS NOTED THAT THE EXECUTION OF EACH JOB REQUIRES A CERTAIN SET OF COMPONENTS OF THE SYSTEM. THIS PAPER PROPOSES A METHOD FOR EVALUATING THE PERFORMANCE OF SUCH A FAULT-TOLERANT COMPUTING SYSTEM FROM THE VEIWPOINT OF JOB EXECUTION.

THE SYSTEM IS FORMULATED INTO A MONOTONE STRUCTURE MODEL, WHERE THE SYSTEM STATE IS DEFINED BY A VECTOR. EACH ELEMENT OF THE VECTOR REPRESENTS THE NUMBER OF AVAILABLE COMPONENTS IN EACH SUBSYSTEM. THE TRANSITION FROM ONE STATE TO ANOTHER IS SPECIFIED ACCORDING TO THE PROBABILITY OF THE OCCURRENCE OF COMPONENT FAILURES. WE ALSO DEFINE THE JOB VECTOR REPRESENTING A SET OF JOBS AND DISCUSS THE MONOTONE STRUCTURE OF JOB SPACE. THIS PAPER CAN BE CONSIDERED AS AN EXTENSION OF THE ANALYSIS OF GRACEFULLY DEGRADABLE SYSTEMS.

PERFORMANCE EVALUATION OF GRACEFULLY DEGRADING SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: JUNE 1979
AUTHORS: GAY, F. KELETSON, M.
SOURCE, PUB. BY, ETC: 9TH ANN INT SYM ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IN CONTRAST TO ULTRARELIABLE SYSTEMS, WHICH USUALLY ACHIEVE A HIGH LEVEL OF PERFORMANCE BY MASKING OUT FAILURES OR BY SWITCHING IN SPARES TO REPLACE FAILED RESOURCES, GRACEFULLY DEGRAGING SYSTEMS ARE DESIGNED TO PROVIDE A HIGH GRADE OF SERVICE BY RECONFIGURING THE SYSTEM AND/OR REALLOCATING RESOURCES WHEN A FAILURE OCCURS. THIS PAPER DESCRIBES AN APPROACH TO EVALUATING THE PERFORMANCE OF SUCH SYSTEMS WHICH COMBINES SYSTEM RELIABILITY AND SYSTEM PERFORMANCE MODELING TECHNIQUES. CAPACITY AND WORKLOAD MODELS BASED ON MARKOV PROCESSES ARE DEVELOPED AND PERFORMANCE MEASURES ARE DEFINED FOR EACH TYPE OF MODEL. THE CAPACITY MODEL CONSIDERS THE EFFECT OF FAILURES ON A SYSTEM'S CAPABILITIES OR "CAPACITY," AS DETERMINED BY THE SYSTEM'S STRUCTURE, WHILE THE WORKLOAD MODEL CONSIDERS THE ABILITY OF A SYSTEM TO SATISFY ITS COMPUTATIONAL DEMANDS. PERFORMANCE MEASURES DERIVED FROM THE CAPACITY MODEL INDICATE CHARACTERISTICS SUCH AS THE PROPORTION OF TIME SPENT IN VARIOUS MODES OF REDUCED CAPACITY, WHILE PERFORMANCE MEASURES DERIVED FROM THE WORKLOAD MODELS

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

PERFORMANCE EVALUATION OF GRACEFULLY DEGRADING SYSTEMS * CONTINUED *

TAKE INTO ACCOUNT THE THROUGHPUT OF THE SYSTEM IN DEGRADED MODES.

PERFORMANCE MODELING OF DISTRIBUTED SYSTEMS

DOC. TYPE: PAPER
AUTHORS: SYKES, D. J.
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER DESCRIBES MODELING TECHNIQUES FOR THE ANALYSIS AND PERFORMANCE PREDICTION OF DISTRIBUTED SYSTEMS. THE TYPE OF SYSTEM BEING MODELED IS COMPRISED OF SEVERAL APPLICATION PROCESSORS AND NETWORK PROCESSORS DISTRIBUTED OVER A LARGE GEOGRAPHIC AREA. THE MODEL DESIGN IS BASED ON A HIERARCHY WHICH DECOMPOSES THE TOTAL PROBLEM INTO SEVERAL SMALLER PROBLEMS. A GLOBAL NETWORK MODEL FORMS THE UPPER LEVEL OF THE HIERARCHY AND REPRESENTS THE INTERNODAL MESSAGE FLOW. THE LOWER LEVEL CONSISTS OF MODELS OF THE INTERNAL OPERATION OF THE INDIVIDUAL NODES. THE MODELS CAN BE IMPLEMENTED USING BOTH ANALYTIC AND DISCRETE SIMULATION METHODS.

PERFORMANCE-RELATED RELIABILITY MEASURES FOR COMPUTING

DOC. TYPE: PAPER
DOC. DATE: JUNE 1978
AUTHORS: BEAUDRY, M.
SOURCE, PUB. BY, ETC: IEEE TRANS ON COMPUTERS VOL C-27, NO.6
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
WE HAVE DEVELOPED MEASURES WHICH REFLECT THE INTERACTION BETWEEN THE RELIABILITY AND THE PERFORMANCE CHARACTERISTICS OF COMPUTING SYSTEMS. THESE MEASURES CAN BE USED TO EVALUATE TRADITIONAL COMPUTER ARCHITECTURES, SUCH AS UNIPROCESSORS AND STANDBY REDUNDANT SYSTEMS; GRACEFULLY DEGRADING SYSTEMS, SUCH AS MULTIPROCESSORS, WHICH CAN REACT TO A DETECTED FAILURE BY RECONFIGURING TO A STATE WITH A DECREASED LEVEL OF PERFORMANCE; AND DISTRIBUTED SYSTEMS. THIS ANALYSIS METHOD, WHICH PROVIDES QUANTITATIVE INFORMATION ABOUT THE TRADEOFFS BETWEEN RELIABILITY AND PERFORMANCE, IS DEMONSTRATED IN SEVERAL EXAMPLES.

PRACTICAL SOFTWARE FAULT TOLERANCE FOR REAL-TIME SYSTEMS. A CLASSIFICATION SYSTEM FOR ERRORS

DOC. TYPE: FINAL REPORT
DOC. DATE: JUNE 1981
AUTHORS: ANDERSON, T. KNIGHT, J.
DOCUMENT NUMBER: TIC# QA 76.5.I58 1982
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
REAL-TIME SYSTEMS OFTEN HAVE VERY HIGH RELIABILITY REQUIREMENTS AND ARE THEREFORE PRIME CANDIDATES FOR THE INCLUSION OF FAULT-TOLERANCE TECHNIQUES. IN ORDER TO PROVIDE TOLERANCE TO SOFTWARE FAULTS, SOME FORM OF STATE RESTORATION IS USUALLY ADVOCATED AS A MEANS OF RECOVERY. STATE RESTORATION CAN BE EXPENSIVE AND THE COST IS EXACERBATED FOR SYSTEMS WHICH UTILIZE CONCURRENT PROCESSES. THE CONCURRENCY PRESENT IN MOST REAL-TIME SYSTEMS AND THE FURTHER DIFFICULTIES INTRODUCED BY TIMING CONSTRAINTS IMPLY THAT PROVIDING TOLERANCE FOR SOFTWARE FAULTS MAY BE INORDINATELY EXPENSIVE OR COMPLEX. THE PAPER ASSERTS THAT THIS IS NOT THE CASE, AND PROPOSES A STRAIGHTFORWARD PRAGMATIC APPROACH TO SOFTWARE FAULT TOLERANCE WHICH IS BELIEVED TO BE APPLICABLE TO MANY REAL-TIME SYSTEMS. THE APPROACH TAKES ADVANTAGE OF THE STRUCTURE OF REAL-TIME SYSTEMS TO SIMPLIFY ERROR RECOVERY, AND A CLASSIFICATION SCHEME FOR ERRORS IS INTRODUCED. RESPONSES TO EACH TYPE OF ERROR ARE PROPOSED WHICH ALLOW SERVICE TO BE MAINTAINED.

PRELIMINARY DRAFT OF VERIFICATION AND VALIDATION GUIDELINES FOR MK-6 TEST STATIONS SOFTWARE

DOC. TYPE: REPORT
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THESE VERIFICATION AND VALIDATION (V&V) GUIDELINES ARE WRITTEN IN A GENERALIZED FASHION. THEY ARE INTENDED TO SERVE AS THE BASIS FROM WHICH INDIVIDUALIZED V&V PLANS WILL BE GENERATED FOR EACH MK-6 TEST STATION. THE OBJECTIVE OF THE V&V GUIDELINES IS TO DEFINE AND DESCRIBE A SET OF V&V TASKS WHICH WILL OVERLAY THE SOFTWARE DEVELOPMENT PROCESS.

PRINCIPLES OF PROGRAM DESIGN

DOC. TYPE: BOOK
DOC. DATE: 1975
AUTHORS: JACKSON, M. A.
SOURCE, PUB. BY, ETC: ACADEMIC PRESS, NY
FILED BY (NAME): SZULEWSKI, P.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

PROBABILITY AND STATISTICS WITH RELIABILITY, QUEUING AND COMPUTER SCIENCE APPLICATIONS
DOC. TYPE: BOOK
DOC. DATE: 1982
AUTHORS: TRIVEDI, K.
SOURCE, PUB. BY, ETC: PRENTICE-HALL
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE AIM OF THIS BOOK IS TO PROVIDE AN INTRODUCTION TO PROBABILITY, STOCHASTIC PROCESSES, AND STATISTICS FOR STUDENTS OF COMPUTER SCIENCE, ELECTRICAL/COMPUTER ENGINEERING, RELIABILITY ENGINEERING, AND APPLIED MATHEMATICS. THE PREREQUISITES ARE TWO SEMESTERS OF CALCULUS, A COURSE ON INTRODUCTION TO COMPUTER PROGRAMMING, AND PREFERABLY, A COURSE ON COMPUTER ORGANIZATION.

I HAVE FOUND THAT THE MATERIAL IN THE BOOK CAN BE COVERED IN A TWO-SEMESTER OR THREE-QUARTER COURSE. HOWEVER, THROUGH A CHOICE OF TOPICS, SHORTER COURSES CAN ALSO BE ORGANIZED. I HAVE TAUGHT THE MATERIAL IN THIS BOOK TO SENIORS AND FIRST-YEAR GRADUATE STUDENTS BUT WITH THE TEXT IN PRINTED FORM, IT COULD BE GIVEN TO JUNIORS AS WELL.

PROBLEMS WITH THE MULTITASKING FACILITIES IN THE ADA PROGRAMMING LANGUAGE
DOC. TYPE: TECH NOTE
DOC. DATE: MAY 1981
AUTHORS: ZUCKERMAN, SUSAN LANA
SOURCE, PUB. BY, ETC: DEFENSE COMMUNICATIONS ENGINEERING CENTER
DOCUMENT NUMBER: TECH NOTE 16-81
FILED BY (NAME): KNOSP, A. A.

PROGRAM MANAGEMENT OF HIGH TECHNOLOGY SOFTWARE AND HARDWARE FOR THE PROPOSED SPACE STATION
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1983
AUTHORS: CHEVERS, E. S.
SOURCE, PUB. BY, ETC: AIAA COMPUTERS IN AEROSPACE IV CONFERENCE
DOCUMENT NUMBER: AIAA-83-2337-CP (NOT INCL IN CONF PROC'GS)
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE NEXT BIG NASA PROJECT COULD BE A SPACE STATION WITH AN INITIAL LAUNCH DATE IN THE 1990-92 TIME FRAME. INCLUDING PRELIMINARY DESIGN, FINAL DESIGN, FABRICATION, BUILDUP AND A TYPICAL 15 YEAR LIFETIME, THE SPACE STATION PROJECT MAY SPAN MORE THAN 35 YEARS. THE POTENTIAL ADVANCES IN TECHNOLOGY WHICH WILL BE MADE DURING THIS TIME ARE VIRTUALLY INCONCEIVABLE. HOWEVER, THE EARLY DATA SYSTEM DESIGN MUST BE FLEXIBLE ENOUGH TO ADAPT TO CHANGING REQUIREMENTS THROUGHOUT THIS PERIOD. DUE TO THE LONG LIFE AND EVOLUTIONARY GROWTH ANTICIPATED, THE HARDWARE AND SOFTWARE DESIGN DECISIONS MADE AT THE BEGINNING OF THE PROJECT MUST CONTINUE TO BE VIABLE FOR MANY YEARS. NEW TOOLS AND LANGUAGES FOR SOFTWARE DEVELOPMENT WILL BE REQUIRED. LONG TERM MAINTAINABILITY OF THE FLIGHT SYSTEM IMPOSES SOME HARDWARE DESIGN LIMITATIONS. POTENTIAL DESIGNS FOR THE SPACE STATION DATA SYSTEM AND TECHNICAL AND MANAGEMENT IMPACTS OF EACH ARE ADDRESSED. NO DECISIONS ARE PRESENTED, JUST THOUGHTS FOR CONSIDERATION AS PRELIMINARY DESIGN STUDIES ARE BEING CONDUCTED.

THE AUTHOR IS WITH THE NASA JOHNSON SPACE CENTER, HOUSTON, TX.

PROGRAMMING EMBEDDED SYSTEMS WITH ADA
DOC. TYPE: BOOK
DOC. DATE: 1982
AUTHORS: DOWNES, V.A. GOLDSACK, S.J.
SOURCE, PUB. BY, ETC: PRENTICE-HALL INTERNATIONAL
DOCUMENT NUMBER: ISBN 0-13-730010-7
FILED BY (NAME): KNOSP, A.
ABSTRACT:
THE PROGRAMMING LANGUAGE ADA HAS BEEN DEVELOPED BY THE UNITED STATES DEPARTMENT OF DEFENSE (DOD) AS PART OF A PROGRAM AIMED AT REDUCING THE HIGH COST OF SOFTWARE FOR EMBEDDED COMPUTER SYSTEMS. THE DOMAIN OF EMBEDDED SYSTEMS COVERS THOSE APPLICATION AREAS WHERE THE SOFTWARE INTERACTS DIRECTLY WITH ELECTROMECHANICAL DEVICES OTHER THAN THE HARDWARE COMPONENTS OF A COMPUTER SYSTEM. SUCH SYSTEMS REQUIRE DIFFERENT PROGRAMMING TECHNIQUES FROM THOSE OF TRADITIONAL DATA PROCESSING, PARTICULARLY WITH RESPECT TO REAL-TIME CONSTRAINTS, INTERACTIONS WITH HARDWARE, ERROR RECOVERY AND CONCURRENT PROCESSING.

ADA HAS BEEN DESIGNED PARTICULARLY FOR USE IN THE IMPLEMENTATION OF EMBEDDED SOFTWARE AND CONTAINS FEATURES AIMED AT FACILITATING THE CONSTRUCTION OF RELIABLE PROGRAMS IN THIS FIELD. HOWEVER, IT IS ALSO A POWERFUL GENERAL PURPOSE HIGH LEVEL LANGUAGE WHICH WILL FIND MANY USES OUTSIDE ITS MAIN APPLICATION DOMAIN. NOT ONLY IS ADA EXPECTED TO OVERTAKE FORTRAN BY THE 1990'S {1} BUT IT ALSO HAS THE POTENTIAL FOR EXTENSION INTO THE DATA PROCESSING STRONGHOLD OF COBOL {2}.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

PROGRAMMING LANGUAGE CONCEPTS
DOC. TYPE: BOOK
DOC. DATE: 1982
AUTHORS: GHEZZI, CARLO JAZAYERI, MEHDI
SOURCE, PUB. BY, ETC: JOHN WILEY & SONS, INC.
FILED BY (NAME): KNOSP, A. A.
ABSTRACT:
THIS BOOK INTRODUCES, ANALYZES, AND EVALUATES THE IMPORTANT CONCEPTS FOUND IN CURRENT PROGRAMMING LANGUAGES.

THIS BOOK IS NOT AN INTRODUCTION TO ANY ONE PROGRAMMING LANGUAGE. IT IS NOT A FEATURE-BY-FEATURE EXAMINATION OF PROGRAMMING LANGUAGES. RATHER, IT PRESENTS AND EVALUTATES CONCEPTS COMMON TO PROGRAMMING LANGUGAES AND THOSE LIKELY TO DICTATE THE EVOLUTION OF PROGRAMMING LANGUAGES IN THE FUTURE.

THIS BOOK WAS DESIGNED TO BE USED AS A TEXTBOOK IN A COURSE ON PROGRAMMING LANGUAGES.

PREREQUISITES FOR THE BOOK ARE FLUENCY IN ONE PROGRAMMING LANGUAGE AND WORKING KNOWLEDGE OF ANOTHER; PREFERABLY, ONE OF THE TWO SHOULD BE BLOCK-STRUCTURED. READING KNOWLEDGE OF PASCAL IS HELPFUL BUT NOT ESSENTIAL. THE READER SHOULD ALSO HAVE SOME SOFTWARE DEVELOPMENT EXPERIENCE.

PARTS OF THE BOOK HAVE BEEN USED OVER THE LAST FIVE YEARS IN A BEGINNING GRADUATE-LEVEL COURSE ON PROGRAMMING LANGUAGES.

PROPOSED REQUIREMENTS METHODOLOGY FOR INTEGRATED AVIONICS SYSTEMS
DOC. TYPE: CSDL REPORT
DOC. DATE: JULY 28, 1983
AUTHORS: BROCK, LARRY DEWOLF, J. BARTON
SOURCE, PUB. BY, ETC: CSDL
DOCUMENT NUMBER: R-1656
FILED BY (NAME): BROCK, LARRY
ABSTRACT:
A METHODOLOGY IS PROPOSED FOR DEFINING THE REQUIREMENTS FOR INTEGRATED AVIONICS SYSTEMS THAT WILL ALLOW MORE EFFECTIVE APPLICATION OF ADVANCED TECHNOLOGY. THE METHODOLOGY SEPARATES THE REQUIREMENTS FROM IMPLIED DESIGNS THAT ARE OFTEN BASED ON OLDER TECHNOLOGIES. THE METHODOLOGY DEFINES THREE BASIC STEPS: DEFINITION OF FUNCTIONAL REQUIREMENTS, THE FUNCTIONAL DESIGN, AND THE IMPLEMENTATION DESIGN. THESE THREE STEPS ARE PERFORMED AT EACH LEVEL IN THE SYSTEM DEVELOPMENT PROCESS AS THE DESIGN IS REFINED FROM HIGH LEVEL MISSION NEEDS DOWN TO THE BASIC COMPONENT LEVEL. THE DESCRIPTION OF THE FUNCTIONAL DESIGN IS BASED ON STRUCTURED ANALYSIS TECHNIQUES AND CONSISTS OF DATA FLOW DIAGRAMS, PROCESS DESCRIPTIONS, AND DATA DESCRIPTIONS. AN EXAMPLE IS PRESENTED THAT ILLUSTRATES POTENTIAL IMPROVEMENTS IN SYSTEM DESIGNS.

PSL/PSA: A COMPUTER AIDED TECHNIQUE FOR STRUCTURED DOCUMENTATION AND ANALYSIS OF INFORMATION PROCESSING
DOC. TYPE: ARTICLE
DOC. DATE: JANUARY 1977
AUTHORS: TEICHROEW, D. HERSHEY, E. A.
SOURCE, PUB. BY, ETC: IEEE TR ON SFTWR ENGRG, SE-3, PP 41-48
FILED BY (NAME): SZULEWSKI, P.

RADIATION DAMAGE IN SINGLE-MODE OPTICAL FIBER WAVEGUIDES
DOC. TYPE: PAPER
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.
SOURCE, PUB. BY, ETC: NAVAL RESEARCH LAB., WASH. DC; PRSNT'D - ?
FILED BY (NAME): KEMP, A.
ABSTRACT:
THE GROWTH AND RECOVERY OF THE RADIATION-INDUCED ATTENUATION AT 0.85 AND 1.3 MICROMETERS HAS BEEN MEASURED IN SINGLE-MODE FIBERS. MODERATE DOSES (ABOUT 1,000 RADS) INDUCE LOSSES THAT ARE AT LEAST AN ORDER OF MAGNITUDE GREATER THAN INTRINSIC, BUT THE RECOVERY OBSERVED IN FIBERS NOT DOPED WITH P MATERIAL IN THE CORE OR CLADDING IS SUFFICIENT TO ELIMINATE THE DAMAGE IN A PERIOD OF 1 TO 24 HOURS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

RADIATION DAMAGE OF OPTICAL FIBER WAVEGUIDES AT LONG WAVELENGTHS
DOC. TYPE: ARTICLE
DOC. DATE: FEBRUARY 1982
AUTHORS: FRIEBELE, E. GINGERICH, M. LONG, K.
SOURCE, PUB. BY, ETC: APPLIED OPTICS, VOL. 21, NO. 3, 1 FEB., 1982
FILED BY (NAME): KEMP, A.
ABSTRACT:
MEASUREMENTS OF THE RADIATION-INDUCED OPTICAL ATTENUATION AT 1.3 MICROMETERS AND THE INDUCED ABSORPTION SPECTRA (0.4-1.7 MICROMETERS) OF STATE-OF-THE-ART PURE SYNTHETIC SILICA AND DOPED SILICA CORE OPTICAL FIBER WAVEGUIDES HAVE BEEN UNDERTAKEN TO CHARACTERIZE THEIR RADIATION RESPONSE AT LONG WAVELENGTHS. IT HAS BEEN OBSERVED THAT RADIATION-INDUCED ABSORPTION BANDS AT LONG WAVELENGTHS CAN GIVE RISE TO SUBSTANTIAL INDUCED LOSSES AT BOTH 1.3 AND 1.55 MICROMETERS IN SOME FIBERS, ESPECIALLY THOSE DOPED WITH P OR B; THE RATIO OF THE DAMAGE AT 1.3 AND 1.55 MICROMETERS TO THAT AT 0.82 MICROMETER IN THESE FIBERS HAS BEEN FOUND TO BE ONLY ABOUT 0.29 AND 0.7L, RESPECTIVELY. IN CONTRAST, PURE FUSED SILICA AND BINARY GE-DOPED SILICA CORE FIBERS HAVE SHOWN THE GREATEST HARDNESS AT LONG WAVELENGTHS. SUGGESTIONS HAVE VEEN MADE FOR THE OPTIMUM WAVELENGTHS AND PREFERRED FIBER COMPOSITIONS TO MINIMIZE THE EFFECTS OF NUCLEAR RADIATION IN FIBER-OPTIC COMMUNICATIONS SYTEMS OPERATING AT LONG WAVELENGTHS.

RADIATION HARDNESS OF NEW TECHNOLOGIES - STATE OF THE ART REVIEW
DOC. TYPE: PAPER
DOC. DATE: NOVEMBER 30, 1982
AUTHORS: LONG. D.
SOURCE, PUB. BY, ETC: GOVT. MICROCIRCUITS APPLICATIONS CONF., 11/82
FILED BY (NAME): KEMP, A.

RECOVERY OF RADIATION DAMAGE IN SINGLE-MODE AND MULTIMODE OPTICAL FIBER WAVEGUIDES
DOC. TYPE: PAPER
DOC. DATE: APRIL 30, 1982
AUTHORS: FRIEBELE, E. LONG, K. GINGERICH, M.
SOURCE, PUB. BY, ETC: TOPICAL MTG., OPTICAL FIBER COMM., PHOENIX, 4/82
FILED BY (NAME): KEMP, A.

REDUNDANT PROGRAMMING IN EUROPE
DOC. TYPE: PAPER
DOC. DATE: JANUARY 1981
AUTHORS: TAYLOR, R.
SOURCE, PUB. BY, ETC: ACM SIGSOFT SEN, V 6, #1, JANUARY 1981
FILED BY (NAME): DEWOLF, J. B.

RELIABILITY AND ACCURACY PREDICTION FOR A REDUNDANT STRAPDOWN NAVIGATOR
DOC. TYPE: ARTICLE
DOC. DATE: SEPTEMBER 1981
AUTHORS: HARRISON, J. V. A. DALY, K. C. GAI, E. G.
SOURCE, PUB. BY, ETC: J. GUIDANCE AND CONTROL, AIAA, VOL.4 NO.5
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
A COMPREHENSIVE APPROACH TO THE EVALUATION OF THE ACCURACY AND RELIABILITY OF A REDUNDANT NAVIGATION SYSTEM IS DESCRIBED. A MARKOV MODEL OF THE REDUNDANT SYSTEM IS USED TO DETERMINE THE PROBABILITIES OF PARTICULAR OPERATIONAL STATE TIME HISTORIES. NAVIGATION SYSTEM ACCURACIES ARE ASSOCIATED WITH THESE STATE TIME HISTORIES THROUGH THE USE OF A MODIFIED COVARIANCE ANALYSIS OF THE SYSTEM'S NAVIGATION ERRORS. SUITABLE SCALAR FIGURES OF MERIT ARE USED TO ASSESS THE IMPACT ON PERFORMANCE OF SIGNIFICANT SYSTEM PARAMETERS. THE ANALYSIS IS APPLIED TO A REDUNDANT NAVIGATOR WHICH IS USED TO TRANSFER A PAYLOAD FROM LAUNCH TO GEOSYNCHRONOUS ORBIT.

RELIABILITY AND AVAILABILITY MODELING OF SYSTEMS FEATURING HARDWARE AND SOFTWARE FAULTS
DOC. TYPE: PAPER
DOC. DATE: JUNE 1977
AUTHORS: LANDROW, H. LAPIE, J.
SOURCE, PUB. BY, ETC: 7TH ANN INT SYM ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
MODELING OF SYSTEMS FEATURING HARDWARE AND SOFTWARE FAULTS IS STUDIED AS A MEANS OF EVALUATING THE AVAILABILITY AND RELIABILITY CHARACTERISTICS. THE CASE OF A NON-REDUNDANT COMPUTER IS STUDIED AND IT IS SHOWN THAT THE UNAVAILABILITY PRESENTS AN OVERSHOOT WITH RESPECT TO ITS ASYMPTOTIC VALUE WHOSE HEIGHT AND LENGTH ARE FUNCTIONS OF THE FAILURE RATES ASSOCIATED WITH THE DIFFERENT DESIGN ERRORS. ALSO, A FAULT-TOLERANT SYSTEM IS STUDIED THAT INCLUDES PROTECTIVE REDUNDANCIES AT THE HARDWARE AND SOFTWARE LEVELS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

RELIABILITY AND AVAILABILITY MODELS FOR FAULT-TOLERANT SYSTEMS

DOC. TYPE: THESIS
DOC. DATE: AUGUST 1983
AUTHORS: LUPPOLD, R.
SOURCE, PUB. BY, ETC: CSDL
DOCUMENT NUMBER: CSDL T-826
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IN THE AREA OF RELIABILITY MODELING, THE FEASIBILITY OF INCORPORATING THE REDUNDANCY MANAGEMENT CHARACTERISTICS INTO A COMBINATORIAL MODEL IS INVESTIGATED. THE REDUNDANCY MANAGEMENT CHARACTERISTICS CONSIDERED INCLUDE: (1) FDI DECISION ERRORS, (2) FDI STRATEGY, AND (3) SYSTEM ARCHITECTURAL EFFECTS. IN EACH OF THESE SITUATIONS, IT IS SHOWN THAT TIME-ORDERED-EVENT SEQUENCES ARE INTRODUCED BY THESE REDUNDANCY MANAGEMENT CHARACTERISTICS WHICH AFFECT THE SYSTEM'S RELIABILITY. THE METHODOLOGY OF INCORPORATING TIME-ORDERED-EVENT SEQUENCES INTO A COMBINATORIAL MODEL IS DEMONSTRATED WITH SEVERAL SIMPLE EXAMPLES. THE RESULTS OF THIS WORK INDICATE THAT THE MARKOV MODELING TECHNIQUE IS VASTLY SUPERIOR TO THE COMBINATORIAL TECHNIQUE FOR DETERMINING THE RELIABILITY OF A FAULT-TOLERANT SYSTEM WITH A DECISION-MAKING REDUNDANCY MANAGEMENT POLICY. SEVERAL TOPICS ARE ADDRESSED IN THE AVAILABILITY MODELING AREA. AMONG THESE ARE: (1) THE EFFECTS OF NONPERFECT COVERAGE ON SYSTEM AVAILABILITY, (2) MODELING PERIODIC PREVENTIVE DIAGNOSTIC TESTS, AND (3) MARKOV MODEL REDUCTION THROUGH STATE AGGREGATION.

RELIABILITY AND AVAILABILITY MODELS FOR MAINTAINED SYSTEMS FEATURING HARDWARE FAILURES AND DESIGN FAULTS

DOC. TYPE: PAPER
DOC. DATE: JUNE 1978
AUTHORS: COSTES, A. LANDRAULT, C. LAPRIE, J.
SOURCE, PUB. BY, ETC: IEEE TRANS ON COMPUTERS VOL C-27 NO.6
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
MODELING OF SYSTEMS FEATURING HARDWARE AND SOFTWARE FAULTS IS STUDIED AS A MEANS OF EVALUATING THE AVAILABILITY AND RELIABILITY CHARACTERISTICS. THE CASE OF A NONREDUNDANT COMPUTER IS STUDIED AND IT IS SHOWN THAT THE UNAVAILABILITY PRESENTS AN OVERSHOOT WITH RESPECT TO ITS ASYMPTOTIC VALUE WHOSE HEIGHT AND LENGTH ARE FUNCTIONS OF THE FAILURE RATES ASSOCIATED WITH THE DIFFERENT DESIGN ERRORS. FAULT-TOLERANT SYSTEMS ARE STUDIED THAT INCLUDE PROTECTIVE REDUNDANCIES BOTH AT THE HARDWARE LEVEL AND AT THE SOFTWARE LEVEL; THE IMPORTANCE OF HOMOGENEOUS SOLUTIONS ON BOTH LEVELS IS SHOWN.

RELIABILITY AND INTEGRITY OF DISTRIBUTED COMPUTING SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: JUNE 24, 1981
AUTHORS: RANDELL, B.
SOURCE, PUB. BY, ETC: 11TH ANL INTNATL SYMP ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
A BRIEF OUTLINE IS PROVIDED OF RECENT WORK IN AN ON-GOING RESEARCH PROJECT SPONSORED BY THE UK SCIENCE RESEARCH COUNCIL.

RELIABILITY INDICES FOR TOPOLOGICAL DESIGN OF COMPUTER COMMUNICATION NETWORKS

DOC. TYPE: PAPER
DOC. DATE: DECEMBER 1981
AUTHORS: SOI, I. AGGARWAL, K.
SOURCE, PUB. BY, ETC: IEEE TRANS ON RELIABILITY VOL R-30 NO.5
DOCUMENT NUMBER: 0018-9529/81/1200-0438
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
SUMMARY & CONCLUSIONS - INCORPORATING NETWORK RELIABILITY PARAMETER IN THE DESIGN OF RELIABLE COMPUTER COMMUNICATION NETWORKS MAKES THE COMPUTATIONS PROHIBITIVE. INTERDEPENDENCE AMONG NETWORK TOPOLOGICAL PARAMETERS DOES NOT PERMIT THE DESIGN OF A MAXIMALLY RELIABLE NETWORK USING ANY ONE OF THE PARAMETERS AND THUS, THERE ARISES A REAL NEED FOR A COMPOSITE RELIABILITY INDEX WHICH GIVES A MORE REALISTIC ASSESSMENT OF NETWORK RELIABILITY. AFTER DISCUSSING EXPERIMENTAL RESULTS REGARDING THE EFFECTS OF VARIOUS TOPOLOGICAL PARAMETERS ON NETWORK RELIABILITY, WE PRESENT TWO HEURISTIC RELIABILITY INDICES WHICH GIVE A FAIR INDICATION OF OVERALL RELIABILITY. A DESIGN PROCEDURE FOR RELIABLE COMPUTER COMMUNICATION NETWORKS BASED ON LOCAL SEARCH TECHNIQUE INCORPORATING THESE RELIABILITY INDICES IS SUGGESTED. HAVING ONLY ONE COMPOSITE RELIABILITY INDEX WHICH IS VERY SIMPLE TO EVALUATE SAVES COMPUTATION WHILE DESIGNING MAXIMALLY RELIABLE COMPUTER NETWORKS AS COMPARED TO THE EXISTING TECHNIQUES BASED ON SEVERAL RELIABLIITY MEASURES.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

RELIABILITY ISSUES IN DISTRIBUTED INFORMATION PROCESSING SYSTEM
DOC. TYPE: PAPER
DOC. DATE: JUNE 1979
AUTHORS: SVOBODOVA, L.
SOURCE, PUB. BY, ETC: 9 ANN INTL SYM ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE GOAL OF THIS RESEARCH IS TO INVESTIGATE THE MATTER OF RELIABILITY IN A CLASS OF DISTRIBUTED INFORMATION PROCESSING SYSTEMS COMPOSED OF HIGHLY AUTONOMOUS NODES CONNECTED BY A COMMUNICATION NETWORK WITH UNPREDICTABLE DELAYS. TWO CLASSES OF ISSUES ARE DISCUSSED: ERROR DETECTION AND REPORTING IN INTERNODE COMMUNICATION AND REPLICATION OF RESOURCES TO ENHANCE AVAILABILITY. ATTENTION IS PAID TO THE PROBLEM OF PERFORMANCE DEGRADATION THAT RESULTS DIRECTLY FROM THE APPLICATION OF THE MECHANISMS NEEDED TO ENSURE RELIABLE OPERATION.

RELIABILITY MODEL DERIVATION OF A FAULT-TOLERANT, DUAL, SPARE-SWITCHING DIGITAL COMPUTER SYSTEM
DOC. TYPE: REPORT
DOC. DATE: MARCH 1974
SOURCE, PUB. BY, ETC: RAYTHEON CO.
DOCUMENT NUMBER: NASA CR-132441
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
A COMPUTER-BASED RELIABILITY PROJECTION AID, TAILORED SPECIFICALLY FOR APPLICATION IN THE DESIGN OF FAULT-TOLERANT COMPUTER SYSTEMS, IS DESCRIBED. ITS MORE PRONOUNCED CHARACTERISTICS INCLUDE THE FACILITY FOR MODELING SYSTEMS WITH TWO DISTINCT OPERATIONAL MODES, MEASURING THE EFFECT OF BOTH PERMANENT AND TRANSIENT FAULTS, AND CALCULATING CONDITIONAL SYSTEM COVERAGE FACTORS. THE UNDERLYING CONCEPTUAL PRINCIPLES, MATHEMATICAL MODELS AND COMPUTER PROGRAM IMPLEMENTATION ARE PRESENTED IN CONSIDERABLE DETAIL.

RELIABILITY MODELING AND ARCHITECTURE OF ULTRA-RELIABLE FAULT-TOLERANT DIGITAL COMPUTERS
DOC. TYPE: PH.D THESIS
DOC. DATE: 1970
AUTHORS: MATHUR, F.
SOURCE, PUB. BY, ETC: UNIV. OF CALIFORNIA, LOS ANGELES
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
ALONG WITH RESEARCH INTO REDUNDANCY TECHNIQUES AND THEIR APPLICATION TO FAULT-TOLERANT COMPUTERS, A SOFTWARE PACKAGE LABELED CARE (COMPUTER AIDED RELIABILITY ESTIMATION) WAS DEVELOPED ON THE UNIVAC 1108 MULTI-PROCESSOR SYSTEM AND IS INTENDED TO SERVE AS A RELIABILITY DESIGN-TOOL TO DESIGNERS OF ULTRA-RELIABLE SYSTEMS BY FACILITATING COMPUTATIONAL RELIABILITY ANALYSIS. CARE IS AN EXTENDABLE PROGRAM AND IS DESIGNED BOTH TO BE INTERACTIVELY ACCESSED FROM REMOTE CONSOLES OR TO BE USED IN BATCH MODE. THE INPUT IS IN THE FORM OF SYSTEM CONFIGURATION SELECTION FOLLOWED BY QUERIES ON THE VARIOUS RELIABILITY PARAMETERS OF INTEREST AND THEIR BEHAVIOR WITH RESPECT TO MISSION TIME, FAILURE RATES, DORMANCY FACTORS, FAILURE COVERAGE, AND ALLOCATION OF SPARES OR DEGREES OF REDUNDANCY.

RELIABILITY MODELS FOR MULTIPROCESSOR SYSTEMS WITH AND WITHOUT PERIODIC MAINTENANCE
DOC. TYPE: PAPER
DOC. DATE: JUNE 1977
AUTHORS: INGLE, A. SIEWIOREK, DANIEL P.
SOURCE, PUB. BY, ETC: 7 ANN INT SYM ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
MULTIPROCESSOR SYSTEMS, ALTHOUGH DESIGNED FOR SPEED AND PROCESSING POWER, LEND INHERENTLY TO REDUNDANCY. APPROPRIATELY DESIGNED DISTRIBUTED INTELLIGENCE SYSTEMS THAT UTILIZE SYSTEM RECONFIGURATION AND GRACEFUL DEGRADATION CAN BE SUBSTANTIALLY MORE RELIABLE THAN UNIPROCESSOR SYSTEMS. RELIABILITY MODELS FOR TWO MULTIPROCESSOR SYSTEMS, C.MMP AND CM*, ARE PRESENTED AND COMPARED WITH LSI-11 PROCESSORS AS BASIS. MOST SYSTEMS ARE SUBJECTED TO PERIODIC MAINTENANCE TO ENSURE PROPER FUNCTIONING. WHEN PERFORMED REGULARLY, THESE INTEGRITY CHECKS ENHANCE CONFIDENCE IN THE SYSTEM, AND ITS EXPECTED MEAN TIME TO FAILURE. EFFECT OF SUCH PERIODIC MAINTENANCE IS MODELED. THE EXPECTED LIFE IS SEEN TO DEPEND STRONGLY ON THE EFFICIENCY OF THE TESTS. THE IMPROVEMENT IN EXPECTED LIFE, HOWEVER, IS OBSERVED TO BE LIMITED BY NON-REDUNDANT PARTS OF A SYSTEM. UNDER PERIODIC MAINTENANCE, CM* SYSTEM OFFERS GREATER LIFE THAN C.MMP FOR TASKS ALLOWING CONSIDERABLE REDUNDANCY.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

RELIABLE SOFTWARE FOR FAULT-TOLERANT SYSTEMS
DOC. TYPE: REPORT
DOC. DATE: APRIL 9, 1982
AUTHORS: DEWOLF, J. BARTON FURTEK, F. C. GAI, E. G.
SOURCE, PUB. BY, ETC: CSDL
DOCUMENT NUMBER: CSDL C-5487
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS REPORT SUMMARIZES WORK WHICH HAS AS ITS OBJECTIVE THE ENHANCEMENT OF SOFTWARE RELIABILITY. A NEW RELIABILITY-IMPROVING TECHNIQUE IS PROPOSED. IN THIS TECHNIQUE, FORMAL SPECIFICATIONS ARE USED TO PROVIDE A REDUNDANT DESCRIPTION OF ASPECTS OF DESIRED SOFTWARE BEHAVIOR. STATIC AND DYNAMIC VERIFICATION ENSURES THAT THE SOFTWARE IMPLEMENTS THE SPECIFICATIONS. DYNAMIC VERIFICATION WOULD BE IMPLEMENTED USING AUTOMATED CODE AUDITING DURING INTEGRATION TESTING AND/OR AT FAULT DETECTION AND DIAGNOSIS, AND AIDS RECOVERY. RELIABLITIY MODELING TECHNIQUES ARE APPLIED TO BOTH HARDWARE AND SOFTWARE TO ASSESS THE ADEQUACY OF THE FAULT-PROTECTION APPROACH.

SELECTING ON-BOARD SATELLITE COMPUTER SYSTEMS
DOC. TYPE: ARTICLE
DOC. DATE: APRIL 1983
AUTHORS: CARNEY, P.
FILED BY (NAME): JOHNSTON, M.
ABSTRACT:
IN THE LAST 20 YRS, SPACE SATELLITE COMPUTERS HAVE EVOLVED FROM RELATIVELY LOW-LEVEL PROGRAMMABLE SEQ LOGIC TO DISTRIBUTED SYSTS OF POWERFUL MINI- AND MICROCOMPUTERS. PRESENT-DAY COMPUTERS ARE RESPONSIBLE NOT ONLY FOR SUBSYSTEM COMPUTATION AND LOGIC BUT ALSO FOR THE MAINTENANCE OF THE SPACECRAFT AND THE SAFE OPERATION OF SATELLITE SUBSYSTEMS AND PAYLOADS.

TODAY, SPACECRAFT DESIGNERS FACE AN EXTREMELY COMPLEX TASK WHEN SELECTING THE SPACECRAFT COMPUTER. THEY MUST ACCURATELY COLLECT AND STATE THEIR OPERATIONAL REQMNTS AND TRADE THESE OFF AGAINST A VARIETY OF COMPUTER SYS IMPLEMENTATIONS. TO PERFORM THIS TASK, THE SPACECRAFT DESIGNER MUST NOT ONLY BE AWARE OF THE COMPUTERS AVAILABLE FROM DIFFERENT VENDORS BUT MUST ALSO BE ABLE TO PROPERLY EVALUATE THEM AGAINST EACH OTHER AND AGAINST HIS REQMNTS. HIS EVALUATION MUST ADDRESS NOT ONLY SUCH QUANTITATIVE PARAMS AS POWER AND WEIGHT BUT ALSO QUALITATIVE FACTORS SUCH AS USABILITY AND DEVELOPMENT RISK. THIS ARTICLE EXPLORES THE PROCESS OF SELECTING A SATELLITE COMPUTER SYSTEM.

DESIGNERS OF FUTURE MISSIONS ARE DEMANDING MORE THAN AN ORDER OF MAGNITUDE INCREASE IN AVAILABLE SATELLITE COMPUTER PROCESSING POWER OVER THE NEXT 10 YEARS.

SIFT - AN ULTRA-RELIABLE AVIONIC COMPUTING SYSTEM
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: MOSES, KURT
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
TIC #AD-A109-274. THE AUTHOR IS WITH BENDIX CORPORATION, FLIGHT SYSTEMS DIVISION, TETERBORO, NEW JERSEY, USA.

SNEAK SOFTWARE ANALYSIS
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: BURATTI, D.L. PINKSTON, W.E. SIMKINS, R.D.
SOURCE, PUB. BY, ETC: THE BOEING CO. - HOUSTON, TX.
FILED BY (NAME): WERNER, R.

SOFTWARE ACQUISITION MANAGEMENT GUIDEBOOK: VERIFICATION
DOC. TYPE: REPORT
DOC. DATE: AUGUST 1977
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THIS REPORT IS ONE OF A SERIES OF SOFTWARE ACQUISITION MANAGEMENT GUIDEBOOKS WHICH PROVIDE INFORMATION AND GUIDANCE FOR ESD PROGRAM OFFICE PERSONNEL WHO ARE CHARGED WITH PLANNING AND MANAGING THE ACQUISITION OF COMMAND, CONTROL, AND COMMUNICATIONS SYSTEM SOFTWARE PROCURED UNDER AIR FORCE 800 SERIES REGULATIONS AND RELATED SOFTWARE ACQUISITION MANAGEMENT CONCEPTS. IT PROVIDES A REVIEW OF THE SOFTWARE VERIFICATION PRACTICES AND PROCEDURES EMPLOYED BY INDUSTRY AND SET FORTH IN RELEVANT DOD AND AIR FORCE REGULATIONS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION
DOC. TYPE: REPORT
DOC. DATE: NOVEMBER 18, 1981
SOURCE, PUB. BY, ETC: CSDL & TIC FILES
DOCUMENT NUMBER: RTCA/DO-178
FILED BY (NAME): WERNER, R.
ABSTRACT:
THE PURPOSE OF THIS DOCUMENT IS TO DESCRIBE TECHNIQUES AND METHODS THAT MAY BE USED FOR THE ORDERLY DEVELOPMENT AND MANAGEMENT OF SOFTWARE FOR AIRBORNE DIGITAL COMPUTER-BASED EQUIPMENT AND SYSTEMS. THE APPLICATION OF THESE TECHNIQUES AND METHODS WILL RESULT IN DOCUMENTED SOFTWARE THAT IS VISIBLE, TESTABLE AND MANAGEABLE -- CHARACTERISTICS THAT ARE IMPORTANT IN MEETING THE QUALITY REQUIREMENTS OF THE USERS AND GOVERNMENT REGULATORY AGENCIES.

SOFTWARE DEVELOPMENT TOOLS
DOC. TYPE: BOOK
DOC. DATE: MARCH 1982
AUTHORS: HOUGHTON, R. C.
SOURCE, PUB. BY, ETC: U. S. DEPT. OF COMMERCE, NATL. BUR. OF STANDARDS
DOCUMENT NUMBER: NBS SPECIAL PUBLICATION 500-88
FILED BY (NAME): SZULEWSKI, P.

SOFTWARE DIVERSITY IN REACTOR PROTECTION SYSTEMS: AN EXPERIMENT
DOC. TYPE: PAPER
DOC. DATE: 1979
AUTHORS: GMEINER, L. VOGES, U.
SOURCE, PUB. BY, ETC: IFAC WORKSHOP SAFECOMP 1979, STUTTGART
FILED BY (NAME): DEWOLF, J. B.

SOFTWARE RELIABILITY - STATUS AND PERSPECTIVES
DOC. TYPE: ARTICLE
DOC. DATE: JULY 1982
AUTHORS: RAMAMOORTHY, C. V. BASTANI, F. B.
SOURCE, PUB. BY, ETC: IEEE TR ON SFTWR ENGRG, V SE-8, #4
DOCUMENT NUMBER: IEEE# 0098-5589/82/0700-0354
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
IT IS ESSENTIAL TO ASSESS THE RELIABILITY OF DIGITAL COMPUTER SYSTEMS USED FOR CRITICAL REAL-TIME CONTROL APPLICATIONS (E.G. NUCLEAR POWER PLANT SAFETY CONTROL SYSTEMS). THIS INVOLVES THE ASSESSMENT OF THE DESIGN CORRECTNESS OF THE COMBINED HARDWARE/SOFTWARE SYSTEM AS WELL AS THE RELIABILITY OF THEHARDWARE. IN THIS PAPER WE SURVEY METHODS OF DETERMINING THE DESIGN CORRECTNESS OF SYSTEMS AS APPLIED TO COMPUTER PROGRAMS. AUTOMATED PROGRAM PROVING TECHNIQUES ARE STILL NOT PRACTICAL FOR REALISTIC PROGRAMS. MANUAL PROOFS ARE LENGTHY, TEDIOUS, AND ERROR-PRONE. SOFTWARE RELIABILITY PROVIDES A MEASURE OF CONFIDENCE IN THE OPERATIONAL CORRECTNESS OF THE SOFTWARE. SINCE THE EARLY 1970'S SEVERAL SOFTWARE RELIABILITY MODELS HAVE BEEN PROPOSED. WE CLASSIFY AND DISCUSS THESE MODELS USING THE CONCEPTS OF RESIDUAL ERROR SIZE AND THE TESTING PROCESS USED. WE ALSO DISCUSS METHODS OF ESTIMATING THE CORRECTNESS OF THE PROGRAM AND THE ADEQUACY OF THE SET OF TEST CASES USED. THESE METHODS ARE DIRECTLY APPLICABLE TO ASSESSING THE DESIGN CORRECTNESS OF THE TOTAL INTEGRATED HARDWARE/SOFTWARE SYSTEM WHICH ULTIMATELY COULD INCLUDE LARGE COMPLEX DISTRIBUTED PRO. SYSTEMS.

SOFTWARE RELIABILITY MEASUREMENT - THE STATE OF THE ART
DOC. TYPE: PAPER
DOC. DATE: JUNE 14, 1982
AUTHORS: MUSA, J. D.
SOURCE, PUB. BY, ETC: PROC, EUROCON '82, 14-18 JUNE 1982, COPENHAGEN
DOCUMENT NUMBER: ISBN 0 444 86419 9
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE FIELD OF SOFTWARE RELIABILITY MEASUREMENT IS SURVEYED. NEEDS FOR AND POTENTIAL USES OF THESE MEASURES ARE OUTLINED. A COMPARISON OF SOFTWARE AND HARDWARE RELIABILITY IS MADE, AND SOME BASIC SOFTWARE RELIABILITY CONCEPTS ARE INTRODUCED. A BRIEF HISTORY AND A SUMMARY OF THE LEADING MODELS IS PRESENTED. ONE OF THE LEADING SOFTWARE RELIABILITY MODELS, THE EXECUTION TIME MODEL, IS PRESENTED IN SOME DETAIL. EXAMPLES OF APPLICATION OF THE THEORY ARE GIVEN. THE CURRENT STATE OF THE ART IS ANALYZED.

PROCEEDINGS OF FIFTH EUROPEAN CONFERENCE ON ELECTROTECHNICS - EUROCON '82, COPENHAGEN, DENMARK. PUBLISHED BY NORTH-HOLLAND PUBLISHING COMPANY, AMSTERDAM, NETHERLANDS. SPONSOR: IEEE; CONVENTION NATIONAL SOCIETY OF ELECTRICAL ENGINEERS, WESTERN EUROPE. RELIABILITY IN ELECTRICAL AND ELECTRONIC COMPONENTS AND SYSTEMS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

SOFTWARE STRUCTURE METRICS BASED ON INFORMATION FLOW
DOC. TYPE: ARTICLE
DOC. DATE: SEPTEMBER 1981
AUTHORS: HENRY, S. KAFURA, D.
SOURCE, PUB. BY, ETC: IEEE TR, SFTWR ENGRG, V SE-7, #5
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
STRUCTURED DESIGN METHODOLOGIES PROVIDE A DISCIPLINED AND ORGANIZED GUIDE TO THE CONSTRUCTION OF SOFTWARE SYSTEMS. HOWEVER, WHILE THE METHODOLOGY STRUCTURES AND DOCUMENTS THE POINTS AT WHICH DESIGN DECISIONS ARE MADE, IT DOES NOT PROVIDE A SPECIFIC, QUANTITATIVE BASIS FOR MAKING THESE DECISIONS. TYPICALLY, THE DESIGNERS' ONLY GUIDELINES ARE QUALITATIVE, PERHAPS EVEN VAGUE PRINCIPLES SUCH AS "FUNCTIONALITY," "DATA TRANSPARENCY," OR "CLARITY." THIS PAPER, LIKE SEVERAL RECENT PUBLICATIONS, DEFINES AND VALIDATES A SET OF SOFTWARE METRICS WHICH ARE APPROPRIATE FOR EVALUATING THE STRUCTURE OF LARGE-SCALE SYSTEMS. THESE METRICS ARE BASED ON THE MEASUREMENT OF INFORMATION FLOW BETWEEN SYSTEM COMPONENTS. SPECIFIC METRICS ARE DEFINED FOR PROCEDURE COMPLEXITY, MODULE COMPLEXITY, AND MODULE COUPLING. THE VALIDATION, USING THE SOURCE CODE FOR THE UNIX OPERATING SYSTEM, SHOWS THAT THE COMPLEXITY MEASURES ARE STRONGLY CORRELATED WITH THE OCCURRENCE OF CHANGES. FURTHER, THE METRICS FOR PROCEDURES AND MODULES CAN BE INTERPRETED TO REVEAL VARIOUS TYPES OF STRUCTURAL FLAWS IN THE DESIGN AND IMPLEMENTATION.

SOME EXPERIENCES OF IMPLEMENTING THE ADA CONCURRENCY FACILITIES ON A DISTRIBUTED MULTIPROCESSOR COMPUTER SYSTEM
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1982
AUTHORS: SHOJA, G. C. HALSALL, F. GRIMSDALE, R. L.
SOURCE, PUB. BY, ETC: SOFTWARE & MICROSYSTEMS, V 1, #6, OCTOBER 1982
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE PAPER FIRST IDENTIFIES THE CONCURRENCY REQUIREMENTS FOR A HIGH-LEVEL CONCURRENT PROGRAMMING LANGUAGE, AND THEN DISCUSSES HOW THE CONCURRENCY PRIMITIVES PROPOSED FOR THE PROGRAMMING LANGUAGE ADA MEET THESE REQUIREMENTS. THIS IS FOLLOWED BY A DESCRIPTION OF AN ACTUAL IMPLEMENTATION OF THE ADA TASKING CONSTRUCTS ON A DISTRIBUTED MULTIPROCESSOR COMPUTER SYSTEM COMPARING A NUMBER OF MULTIPROCESSOR SHARED MEMORY STATIONS LINKED TO A SERIAL COMMUNICATIONS NETWORK. A BRIEF DESCRIPTION OF THE SYSTEM AND THE OVERHEADS INVOLVED WITH IMPLEMENTING THE RENDEZVOUS MECHANISM ON THIS ARCHITECTURE IS PRESENTED AND ALSO THE IMPLICATIONS OF PROCESSING INTERRUPTS BY MAPPING THEM INTO EXTERNAL ENTRY CALLS TO HIGH LEVEL SERVICE TASKS ARE DISCUSSED.

THE AUTHORS ARE WITH THE SCHOOL OF ENGINEERING & APPLIED SCIENCES, UNIVERSITY OF SUSSEX, FALMER, BRIGHTON BN1 9QT, ENGLAND.

SPACE PROCESSOR TECHNOLOGY TRADEOFFS
DOC. TYPE: PAPER
DOC. DATE: 1982
AUTHORS: CRANE, W. W. CHAN, W. S. HUANG, C. C.
SOURCE, PUB. BY, ETC: SPIE, V 319, VHSIC TECHN FOR ELECTRO-OPTIC APPLNS
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
COMPUTATIONAL REQUIREMENTS FOR PROPOSED SATELLITE SURVEILLANCE MISSIONS HAVE GROWN BEYOND THE CAPABILITIES OF EXISTING SPACE QUALIFIED COMPUTERS. IN FACT, THROUGHPUT FOR A SPACE PROCESSOR OF THE FUTURE IS EXPECTED TO BE IN EXCESS OF 50 MILLION INSTRUCTIONS PER SECOND (MIPS). THIS PROCESSING CAPABILITY MUST BE ATTAINED WITH MINIMUM SYSTEM SIZE, WEIGHT AND POWER. RECENT STUDIES INDICATE THAT A DISTRIBUTED PROCESSING SYSTEM IS A REALISTIC METHOD OF ATTAINING THIS GOAL. COMPLEMENTARY METAL OXIDE SEMICONDUCTOR (CMOS) AND VERY HIGH SPEED INTEGRATED CIRCUITS (VHSIC) TECHNOLOGIES WERE EVALUATED FOR THE DESIGN OF A MODULAR, HIGH THROUGHPUT, SPACE COMPUTER. OPERATIONAL AND PHYSICAL PROPERTY ATTRIBUTES CONSIDERED INCLUDE: 1) PROCESSING THROUGHPUT VERSUS SYSTEM POWER, 2) LOCAL HEAT DISSIPATION, 3) DEVICE RADIATION TOLERANCE AND 4) MULTIPROCESSOR COMMUNICATION TECHNIQUES. THE CONCLUSION OF THE EVALUATION MAKES A SERIES OF RECOMMENDATIONS AND SUGGESTS A SET OF TESTS TO ENABLE THE SPACE COMMUNITY TO SELECT THE OPTIMUM TECHNOLOGY AND SYSTEM ARCHITECTURE FOR A MODULAR SPACE PROCESSING SYSTEM.

SPACE STATION INFORMATION SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: SWINGLE, W. J. MCKAY, C. W.
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE SPACE OPERATIONS INFORMATION SYSTEM IS DEFINED AND CHARACTERIZED IN A WIDE PERSPECTIVE. INTERACTIVE SUBSETS OF THE TOTAL SYSTEM ARE DEFINED AND DISCUSSED. PARTICULAR ATTENTION IS PAID TO THE CONCEPT OF END-TO-END SYSTEMS AND THEIR REPETITIVE POPULATION WITHIN THE TOTAL SYSTEM. HIGH LEVEL PROGRAM GOALS ARE REVIEWED AND RELATED TO MORE EXPLICIT SYSTEM REQUIREMENTS AND USER NEEDS. EMPHASIS IS PLACED ON THE UTILITY AND COST EFFECTIVENESS OF DATA SYSTEM SERVICES FROM A USER STANDPOINT. PRODUCTIVITY, AS A QUANTITATIVE GOALS, IN BOTH DEVELOPMENT AND OPERATIONAL PHASES IS ALSO ADDRESSED. CRITICAL

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

SPACE STATION INFORMATION SYSTEMS * CONTINUED *

ASPECTS OF THE APPROACH TO SUCCESSFUL DEVELOPMENT OF THE DATA MANAGEMENT SYSTEM ARE DISCUSSED ALONG WITH RECOMMENDATIONS IMPORTANT TO ADVANCED DEVELOPMENT ACTIVITIES. CURRENT AND PLANNED ACTIVITY IN BOTH TECHNOLOGY AND ADVANCED DEVELOPMENT AREAS ARE REVIEWED WITH EMPHASIS ON THEIR IMPORTANCETO PROGRAM SUCCESS.

SPACECRAFT COMPUTERS: STATE-OF-THE-ART SURVEY

DOC. TYPE: ARTICLE
DOC. DATE: APRIL 1983 .
AUTHORS: THEIS, D.
FILED BY (NAME): JOHNSTON, M.
ABSTRACT:

PUTTING COMPUTERS INTO SPACECRAFT SYSTS OFFERS MANY POTENTIAL ADVANTAGES WHICH ARE ABOUT TO INCREASE DRAMATICALLY.

MOST SATELLITE EQUIP-ANTENNAS, SOLAR CELLS, ETC., REPRESENT RELATIVELY STABLE TECHNOLOGIES. ONBOARD COMPUTING, HOWEVER, IS IN A DYNAMIC STAGE. THIS ARTICLE SURVEYS THE STATE OF THE ART.

THE FIRST PART COVERS SPACE-QUALIFIED GENERAL-PURPOSE, OR GP, FLT COMPUTERS. THESE ARE USED MOSTLY IN AIR FORCE SATELLITE SYSTS FOR HOUSEKEEPING FUNC SUCH AS ATT AND VEL CNTL, THERMAL MGMT, AND MGMT OF POWER AND OTHER RESOURCES.

THE SURVEY INCL NEWER GP FLT COMPUTERS THAT ARE BEING DEVELOPED OR ARE READY FOR QUALIFICATION. THE CPU DESIGNS OF THESE NEWER UNITS INCORPORATE EITHER BIPOLAR BIT-SLICE OR CUSTOM MOS CHIPS.

THE FIRST SECTION OF THE ARTICLE CONCLUDES WITH A BRIEF DISCUSSION OF RAM USE IN SPACECRAFT COMPUTERS.

THE SECOND SECTION COVERS PAYLOAD PROCESSING COMPUTERS. THE INCREASING AVAILABILITY OF CHIPS SUITABLE FOR IMPLEMENTING POWERFUL ONBOARD PROCESSING SYSTS IS MAKING PAYLOAD PROCESSING MORE PRACTICAL.

THE THIRD AND FINAL SECTION SURVEYS RELATED TECHNOLOGY IN SPACECRAFT COMPUTERS, INCLUDING THE IMPACT OF VHSIC ON SPACECRAFT COMPUTER SYSTS AND MASS STORAGE UNITS.

SPECIAL ISSUE (1ST) ON FAULT-TOLERANT COMPUTING

DOC. TYPE: JOURNAL
DOC. DATE: NOVEMBER 1971
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-20, #11
FILED BY (NAME): LALA, J. H.

SPECIAL ISSUE (2ND) ON FAULT-TOLERANT COMPUTING

DOC. TYPE: JOURNAL
DOC. DATE: MARCH 1983
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-22, #3
FILED BY (NAME): LALA, J. H.

SPECIAL ISSUE (3RD) ON FAULT-TOLERANT COMPUTING

DOC. TYPE: JOURNAL
DOC. DATE: JULY 1974
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-23 #7
FILED BY (NAME): LALA, J. H.

SPECIAL ISSUE (4TH) ON FAULT-TOLERANT COMPUTING

DOC. TYPE: JOURNAL
DOC. DATE: MAY 1975
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-24, #5
FILED BY (NAME): LALA, J. H.

SPECIAL ISSUE (5TH) ON FAULT-TOLERANT COMPUTING

DOC. TYPE: JOURNAL
DOC. DATE: JUNE 1976
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-25, #6
FILED BY (NAME): LALA, J. H.

SPECIAL ISSUE (6TH) ON FAULT-TOLERANT COMPUTING

DOC. TYPE: JOURNAL
DOC. DATE: JUNE 1978
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-27, #6
FILED BY (NAME): LALA, J. H.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

SPECIAL ISSUE (7TH) ON RELIABLE AND FAULT-TOLERANT COMPUTING
DOC. TYPE: JOURNAL
DOC. DATE: JULY 1982
SOURCE, PUB. BY, ETC: IEEE TR ON COMPUTERS, V C-31, #7
FILED BY (NAME): LALA, J. H.

SPECIFICATION AND ANALYSIS OF COMMUNICATION IN DISTRIBUTED DATA PROCESSING SYSTEMS
DOC. TYPE: ARTICLE
DOC. DATE: APRIL 1982
AUTHORS: ORANEN, J.
SOURCE, PUB. BY, ETC: SAEHKO ELECTR. & ELECTRON. V 55, #4 (APR 1982)
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE POTENTIAL BENEFITS OF DISTRIBUTED DATA PROCESSING SYSTEMS AND COMPUTER NETWORKS HAVE BEEN GENERALLY REALIZED. LESS ATTENTION HAS BEEN PAID TO COMPUTER COMMUNICATION, ALTHOUGH COMMUNICATION IS THE SOLE MEANS OF CONTROL IN DISTRIBUTED DATA PROCESSING SYSTEMS. IN THIS STUDY EMPHASIS ISLAID ON STATE-OF-ART METHODS OF LOGICAL MODELLING, SPECIFICATION AND ANALYSIS OF COMPUTER COMMUNICATION. THAT KNOWLEDGE IS VERY RELEVANT FOR ANYONE HAVING TO DESIGN, ANALYSE OR EVALUATE DISTRIBUTED COMPUTING SYSTEMS.

THE AUTHOR IS WITH THE POSTS AND TELECOMMUNICATIONS OF FINLAND, HELSINKI, FINLAND. THE JOURNAL (SAEHKO ..) IS PUBLISHED IN FINLAND.

SPECIFYING SOFTWARE REQUIREMENTS FOR COMPLEX SYSTEMS: NEW TECHNIQUES AND THEIR APPLICATION
DOC. TYPE: ARTICLE
DOC. DATE: 1980
AUTHORS: HENINGER, K. L. ET AL
SOURCE, PUB. BY, ETC: IEEE TRANS ON SFTWRE ENGRG, V SE-6, NO. 1
FILED BY (NAME): SZULEWSKI, P.

STAGE - STATE RELIABILITY ANALYSIS TECHNIQUE
DOC. TYPE: PAPER
DOC. DATE: JUNE 1981
AUTHORS: STERN, A.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE, 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
CONVENTIONAL RELIABILITY ANALYSIS TECHNIQUES, I.E., FAULT-TREE AND BOOLEAN ALGEBRA METHODS ARE DIFFICULT TO APPLY TO REDUNDANT SYSTS WITH COMPLEX INTERACTIONS & REDUNDANCY MGMT PHILOSOPHIES. SOME ADVANCED FLT CNTL SYSTS, FOR EX: EMPLOY MULTIPLE REDUNDANT CHANS WHICH, WITH PROPER REDUNDANCY MGMT & FAILURE DETECTION, CAN DEGRADE TO SIMPLEX OPER. THE RELIABILITY ANAL MUST PROPERLY ACCT FOR THE DEFINED SUCCESS CRITERIA, REDUNDANCY LEVEL, REDUNDANCY MGMT TECH, SYS DEPENDENCIES, & FAILURE DETECTION COVERAGE. THE STAGE-STATE RELIABILITY ANAL TECH PROPERLY ACCTS FOR THESE FACTORS. IT IS ALSO COMPUTATIONALLY SIMPLE SUCH THAT TRIPLEX REDUNDANT SYSTS HAVE BEEN ANALYZED USING AN EARLY 1970'S DESKTOP COMPUTER.

THIS METHOD IS WELL SUITED FOR ANALYSIS BY THE SYS ARCHITECT. THE PROCESS BEGINS WITH A SYS BLOCK DIAG SHOWING ALL ELEMENT CONNECTIONS. A SUCCESS LOGIC DIAG IS THEN WRITTEN REFLECTING ALL POSSIBLE SUCCESS STATES. THE PROBABILITY OF SUCCESS EQUATION IS WRITTEN DIRECTLY FROM THE LOGIC DIAG & EVAL BY SUBSTITUTING THE PROBABILITY EXPRESSION FOR EACH SYS ELEMENT. MULTIPLE SUCCESS CRITERIA CAN BE APPLIED TO 1 PROB FORMULATION SIMPLY BY DELETING THOSE STATES WHICH DO NOT SATISFY THE CRITERION.

STATE-OF-THE-ART COMPUTER MONITORING EQUIPMENT
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: NELSON, H. G.
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
TIC #AD-A109-274. THE AUTHOR IS WITH THE NAVAL WEAPONS CENTER, FACILITY ENGINEERING BRANCH (CODE 3115), CHINA LAKE, CA 93555 USA.

STRUCTURED ANALYSIS (SA): A LANGUAGE FOR COMMUNICATING IDEAS
DOC. TYPE: ARTICLE
DOC. DATE: JANUARY 1977
AUTHORS: ROSS, D.
SOURCE, PUB. BY, ETC: IEEE TRANS ON SFTWRE ENGR'G, V SE-3, #1, PP 16-23
FILED BY (NAME): SZULEWSKI, P.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

STRUCTURED DESIGN
DOC. TYPE: ARTICLE
DOC. DATE: MAY 1974
AUTHORS: STEVENS, W. P. MEYERS, G. J. CONSTANTINE, L. L.
SOURCE, PUB. BY, ETC: IBM SYSTEMS JOURNAL, V 13, #2, PP 115-139
FILED BY (NAME): SZULEWSKI, P.

SUITABILITY OF ADA TASKS FOR DISTRIBUTED COMPUTING
DOC. TYPE: THESIS
DOC. DATE: SEPTEMBER 1, 1983
AUTHORS: GILBERT, LUCY
SOURCE, PUB. BY, ETC: CSDL & TIC
FILED BY (NAME): KNOSP, ALTON
ABSTRACT:
AS THE SPEED REQUIREMENTS OF COMPUTATION BECOME FASTER AND THE COST OF VARIOUS COMPONENTS BECOMES LOWER, DISTRIBUTED OR PARALLEL PROCESSING PROVIDES AN INCREASINGLY MORE DESIRABLE METHOD FOR SOLVING COMPLEX SYSTEMS. CONCURRENT SOLUTIONS PROVIDED BY THE TASKING CONSTRUCT IN THE PROGRAMMING LANGUAGE ADA ARE A NATURAL WAY OF DECOMPOSING MANY PROBLEMS WHETHER THE IMPLEMENTATION IS ACTUALLY ONE OR MANY PROCESSORS. THIS THESIS EVALUATES THE SUITABILITY OF ADA FOR DISTRIBUTED COMPUTING. SEVERAL EXAMPLES ARE PRESENTED, AND IT IS SHOWN THAT THERE IS A LIMITATION IN THE AMOUNT OF CONCURRENCY WHICH MAY BE OBTAINED FROM USING THE RENDEZVOUS CONCEPT OF ADA. AN ANALYSIS OF THIS PROBLEM IS GIVEN, AND SEVERAL SOLUTIONS ARE PROPOSED. SOME OF THESE SOLUTIONS ARE SUITABLE FOR SPECIFIC APPLICATIONS; OTHERS ARE GENERAL BUT INVOLVE A COST OF PROGRAM COMPLEXITY AND TASK OVERHEAD.

SURF - A PROGRAM FOR DEPENDABILITY EVALUATION OF COMPLEX FAULT-TOLERANT COMPUTING SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: JUNE 1981
AUTHORS: COSTES, A. DOUCET, J. LANDRAULT, C. ET AL
SOURCE, PUB. BY, ETC: 11TH ANN INT SYMP ON FAULT-TOLERANT COMPUTING
DOCUMENT NUMBER: CH1600-6/81/0000/0072
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
SURF IS A MARKOV-BASED PROGRAM DEVOTED TO RELIABILITY AND AVAILABILITY EVALUATION OF COMPLEX FAULT-TOLERANT COMPUTING SYSTEMS. ITS MAIN FEATURES ARE (A) COMPUTER-AIDED CONSTRUCTION OF THE BEHAVIOR MODEL OF A SYSTEM, (B) ACCOUNTING FOR NON-CONSTANT TRANSITION RATES ENABLED THROUGH THE METHOD OF STAGES, AND (C) ANALYTIC PROCESSING OF THE MODEL. THE SPECIFICATION, DESIGN AND REALIZATION OF THE PROGRAM ARE GIVEN, AS WELL AS SOME PERFORMANCE MEASURES (MEMORY CAPACITY, EXECUTION TIME).

AUTHORS ARE: A. COSTES; J. DOUCET; C. LANDRAULT; AND J. LAPRIE.

SURF - A PROGRAM FOR MODELING AND RELIABILITY PREDICTION FOR FAULT-TOLERANTCOMPUTING SYSTEMS
DOC. TYPE: PAPERS
DOC. DATE: 1978
AUTHORS: LANDRAULT, C. LAPRIE, J.
SOURCE, PUB. BY, ETC: PROC OF JERUSALEM CONF ON INFO TECH, 3RD
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE INCREASING COMPLEXITY OF SYSTEMS NECESSITATES THE DEVELOPMENT OF TOOLS OF EVALUATION ENABLING THE DESIGNERS TO MAKE MOTIVATED CHOICES. IN THIS PAPER, WE PRESENT A MODELING AND RELIABILITY ESTIMATION PROGRAM USING STOCHASTIC PROCESSES AND PARTICULARLY MARKOV PROCESSES. IN THE FIRST PART, WE GIVE THE DIFFERENT GOALS FOR THIS TOOL AND WE DESCRIBE THE DIFFERENT AVAILABLE TECHNIQUES FOR RELIABILITY ESTIMATION. THE SECOND PART OF THIS PAPER IS DEVOTED TO DESCRIPTION OF THE SURF PROGRAM. LASTLY, WE GIVE, IN THE THIRD PART, AN EXAMPLE OF AN APPLICATION WHICH ENABLES AN APPRECIATION OF THE POSSIBILITIES OF SUCH AN APPROACH.

SURVIVAL AND DISPATCH PROBABILITY MODELS FOR THE FTMP COMPUTER
DOC. TYPE: PAPER
DOC. DATE: JUNE 1978
AUTHORS: LALA, JAYNARAYAN H. HOPKINS, ALBERT L.
SOURCE, PUB. BY, ETC: 8TH ANN INT SYM ON FAULT-TOLERANT COMPUTING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THE FTMP COMPUTER, ONE OF TWO AIRCRAFT FAULT-TOLERANT ARCHITECTURES ENTERING THE PROTOTYPING PHASE UNDER NASA SPONSORSHIP, IS DESIGNED TO EXHIBIT SYSTEM FAILURE AT A RATE BELOW 10E-9 FAILURES PER HOUR, REQUIRED IN ORDER TO PERFORM SUCH CRITICAL FUNCTIONS AS "FLY-BY-WIRE" ACTIVE CONTROL AND AUTOMATIC LANDINGS IN COMMERCIAL AIRCRAFT. MATHEMATICAL MODELS HAVE BEEN DEVELOPED TO ASSESS THE FAILURE RATE DUE TO RANDOM FAILURES USING ARCHITECTURAL PARAMETERS AND HAZARD RATES AS INDEPENDENT VARIABLES. SPECIFICALLY, THE SYSTEM SURVIVAL PROBABILITY IS COMPUTED AS A FUNCTION OF TIME USING COMBINATORIAL AND MARKOV PROCESS MODELS. EACH MODEL IS APPLICABLE TO A PARTICULAR MODE OF FAILURE. THE RELIABILITY COMPUTATIONS ARE REPEATED FOR THE CASE OF INTERMITTENT FAULTS, USING A MODEL BASED ON A METHOD FIRST PUBLISHED BY BREUER. THE IMPACT OF TRANSITION FREQUENCY AND LATENCY OF INTERMITTENT FAULTS ON THE SYSTEM RELIABILITY IS EVALUATED. FINALLY, THIS PAPER

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

SURVIVAL AND DISPATCH PROBABILITY MODELS FOR THE FTMP COMPUTER * CONTINUED *

ADDRESSES THE QUESTION OF DISPATCH RELIABILITY OF THE MULTIPROCESSOR FOR THE SPECIFIC APPLICATION OF COMMERCIAL AIRLINERS.

SYSTEM DESIGN SELECTION FOR DISTRIBUTED DATA SYSTEMS

DOC. TYPE: PAPER
AUTHORS: HEINSELMAN, R. C.
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:

DISTRIBUTED PROCESSING, THE COOPERATIVE PROCESSING OF USER APPLICATIONS BY SEPARATELY LOCATED COMPUTER SYSTEMS, IS A COMPUTER DESIGN PHILOSOPHY THAT IS QUITE POPULAR TODAY. HOWEVER, DESPITE MANY ADVANTAGES, DISTRIBUTED PROCESSING IS NOT SUITABLE FOR ALL ORGANIZATIONS REQUIRING DATA PROCESSING. IN ORDER FOR A DISTRIBUTED PROCESSING INSTALLATION TO SUCCEED, SEVERAL CONDITIONS MUST BE MET. THE CORPORATE NEEDS FOR CONTROL, PERSONNEL DISTRIBUTION AND SKILL LEVELS, AND THE AUTOMATION OF MANUAL TASKS MUST BE SATISFIED BY THE SYSTEM DESIGN. ONCE IT IS ASCERTAINED THAT CERTAIN DISTRIBUTED SYSTEM DESIGNS CAN MEET THIS LEVEL OF CORPORATE REQUIREMENTS, THEN ADDITIONAL CONDITIONS MUST BE CONSIDERED BEFORE A FINAL SELECTION IS MADE. THE CORPORATE REQUIREMENTS FOR SECURITY, INTEGRITY OF DATA, QUERY RESPONSE TIMES, AND SYSTEM AVAILABILITY MUST BE MET USING AVAILABLE AND COST EFFECTIVE PRODUCTS. TOPROPERLY CONSIDER ALL OF THESE MATTERS A COMPREHENSIVE DESIGN STUDY MUST BE PERFORMED. THIS PAPER CONSIDERS A FIVE PHASE DESIGN PROCESS WITH EMPHASIS PLACED ON PERFORMANCE EVALUATION. A CASE STUDY IS PRESENTED AS AN EXAMPLE OF THE APPLICATION OF THIS SYSTEMS DESIGN SELECTION METHODOLOGY.

SYSTEM STRUCTURE FOR SOFTWARE FAULT TOLERANCE

DOC. TYPE: ARTICLE
DOC. DATE: JUNE 1975
AUTHORS: RANDELL, B.
SOURCE, PUB. BY, ETC: ??
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:

THE PAPER PRESENTS, AND DISCUSSES THE RATIONALE BEHIND A METHOD FOR STRUCTURING COMPLEX COMPUTING SYSTEMS BY THE USE OF WHAT WE TERM "RECOVERY BLOCKS", "CONVERSATIONS" AND "FAULT-TOLERANT INTERFACES". THE AIM IS TO FACILITATE THE PROVIDISION OF DEPENDABLE ERROR DETECTION AND RECOVERY FACILITIES WHICH CAN COPE WITH ERRORS CAUSED BY RESIDUAL DESIGN INADEQUACIES, PARTICULARLY IN THE SYSTEM SOFTWARE, RATHER THAN MERELY THE OCCASIONAL MALFUNCTIONING OF HARDWARE COMPONENTS.

TACTICAL DIGITAL SYSTEMS DOCUMENTATION STANDARDS

DOC. TYPE: PAPER
DOC. DATE: AUGUST 8, 1974
SOURCE, PUB. BY, ETC: DEPARTMENT OF THE NAVY
DOCUMENT NUMBER: SECNAVINST 3560.1
FILED BY (NAME): WERNER, R. E.
ABSTRACT:

THE PURPOSE OF THIS DOCUMENT IS TO IDENTIFY, NAME, AND DESCRIBE THAT SET OF DOCUMENTS NECESSARY TO SUPPORT THE DEVELOPMENT AND MAINTENANCE OF DIGITAL PROCESSOR PROGRAMS FOR COMBAT SYSTEMS. IT IS INTENDED THAT THIS DOCUMENT BE THE VEHICLE BY WHICH THE PROCURING AGENCY SHALL SELECT THE NECESSARY DOCUMENTATION FOR PROJECT DEVELOPMENT.

TASK ALLOCATION IN FAULT-TOLERANT DISTRIBUTED SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: MARCH 1983
AUTHORS: BANNISTER, J. TRIVEDI, K.
SOURCE, PUB. BY, ETC: ACTA INFORMATION
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:

THIS PAPER EXAMINES TASK ALLOCATION IN FAULT-TOLERANT DISTRIBUTED SYSTEMS. THE PROBLEM IS FORMULATED AS A CONSTRAINED SUM OF SQUARES MINIMIZATION PROBLEM. THE COMPUTATIONAL COMPLEXITY OF THIS PROBLEM PROMPTS US TO CONSIDER AN EFFICIENT APPROXIMATION ALGORITHM. WE SHOW THAT THE RATIO OF THE PERFORMANCE OF THE APPROXIMATION ALGORITHM TO THAT OF THE OPTIMAL SOLUTION IS BOUNDED BY 9M/(8(M-R+1)), WHERE M IS THE NUMBER OF PROCESSORS TO BE ALLOCATED AND R IS THE NUMBER OF TIMES EACH TASK IS TO BE REPLICATED. EXPERIENCE WITH THE ALGORITHM SUGGESTS THAT EVEN BETTER PERFORMANCE RATIOS CAN BE EXPECTED.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

TECHNICAL PROGRAM MANAGEMENT REQUIREMENTS FOR NAVY STRATEGIC SYSTEMS PROJECT OFFICE ACQUISITIONS
DOC. TYPE: ARTICLE
DOC. DATE: JULY 14, 1981
SOURCE, PUB. BY, ETC: STRATEGIC SYSTEMS PROJECTS
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THIS SPECIFICATION STATES THE REQUIREMENTS FOR TECHNICAL PROGRAM MANAGEMENT APPLICABLE TO NAVY STRATEGIC SYSTEMS PROJECT OFFICE PRIME CONTRACTORS AND SELECTED SUBCONTRACTORS. THE PRIMARY PURPOSE IS TO DEFINE THE MANAGEMENT ACTIONS AND TECHNICAL DISCIPLINES TO BE INVOKED ON CONTRACTORS TO ASSURE ACQUISITION OF A WEAPON SYSTEM THAT WILL ACHIEVE OPERATIONAL REQUIREMENTS. THESE MANAGEMENT ACTIONS AND TECHNICAL DISCIPLINES INCLUDE PROGRAM MANAGEMENT, DESIGN CONTROL, AND TEST AND MEASURING EQUIPMENT AND STANDARDS CONTROL. IT ENCOMPASSES REQUIREMENTS FOR MANAGEMENT OF THESE TECHNICAL DISCIPLINES WHICH ARE CONSISTENT WITH DEFENSE ACQUISITION REGULATIONS (DARS) AND OTHER APPROPRIATE DEPARTMENT OF DEFENSE DIRECTIVES. IT DELINEATES MANAGEMENT ACTIONS AND DISCIPLINES REQUIRED THROUGHOUT THE PRODUCT LIFE CYCLE.

TECHNOLOGY NEEDS FOR AIR FORCE AUTONOMOUS SPACECRAFT
DOC. TYPE: INT. REPORT
DOC. DATE: MARCH 16, 1982
AUTHORS: SCULL, J.
SOURCE, PUB. BY, ETC: USAF HQ SPACE DIV, LOS ANGELES; PREP: NASA/JPL
DOCUMENT NUMBER: TIC# AD A120492; SD-TR-82-50
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THIS VOLUME PRESENTS AN ASSESSMENT OF THE TECHNOLOGY NEEDS FOR ACHIEVING AUTONOMOUS OPERATION OF AIR FORCE SPACECRAFT. THE TECHNOLOGIES IDENTIFIED WERE SELECTED BASED ON AUTONOMY GOALS, MISSION DRIVERS, AND A TECHNOLOGY FORECAST. THE ASSESSMENT CONCLUDED THAT NEAR TERM AUTONOMY GOALS COULD BE MET BY USING CURRENT TECHNOLOGY SUPPLEMENTED BY CERTAIN KEY DEVELOPMENTS SUCH AS AUTONOMOUS SYSTEM METHODOLOGY, FAULT-TOLERANT COMPUTERS, AUTONOMOUS NAVIGATION, AND RELIABLE MASS DATA STORAGE.

THE AMERICAN HERITAGE DICTIONARY - SECOND COLLEGE EDITION
DOC. TYPE: BOOK
DOC. DATE: 1982
SOURCE, PUB. BY, ETC: HOUGHTON MIFFLIN COMPANY
DOCUMENT NUMBER: ISBN 0-395-32943-4
FILED BY (NAME): KNOSP, A.

THE ANALYSIS OF THE STATISTICAL AND HISTORICAL INFORMATION GATHERED DURING THE DEVELOPMENT OF THE SHUTTLE ORBITER PRIMARY FLIGHT SOFTWARE
DOC. TYPE: FINAL REPORT
DOC. DATE: JUNE 1982
AUTHORS: SIMMONS, D. MARCHBANKS, M. QUICK, M.
SOURCE, PUB. BY, ETC: TEXAS A&M RESEARCH FOUNDATION
DOCUMENT NUMBER: AF-82-17
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
THIS REPORT PRESENTS THE RESULTS OF THE PROJECT FOR "THE ANALYSIS OF THE STATISTICAL AND HOSTORICAL INFORMATION GATHERED DURING THE DEVELOPMENT OF THE SHUTTLE ORBITER PRIMARY FLIGHT SOFTWARE". THIS WORK WAS PERFORMED BY THE TEXAS ENGINEERING EXPERIMENT STATION OF TEXAS A&M UNIVERSITY, COLLEGE STATION, TEXAS. THIS WORK WAS PERFORMED FOR THE NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, JOHNSON SPACE FLIGHT CENTER, HOUSTON, TEXAS, UNDER GRANT NCC9-2.

THE C PROGRAMMING LANGUAGE
DOC. TYPE: BOOK
DOC. DATE: 1978
AUTHORS: KERNIGHAN, BRIAN RITCHIE, DENNIS
SOURCE, PUB. BY, ETC: PRENTICE-HALL INC., ENGLEWOOD CLIFFS, N.J. 07632
DOCUMENT NUMBER: ISBN 0-13-110163-3
FILED BY (NAME): KNOSP, A.
ABSTRACT:
C IS A GENERAL-PURPOSE PROGRAMMING LANGUAGE. IT HAS BEEN CLOSELY ASSOCIATED WITH THE UNIX SYSTEM, SINCE IT WAS DEVELOPED ON THAT SYSTEM, AND SINCE UNIX AND ITS SOFTWARE ARE WRITTEN IN C. THE LANGUAGE, HOWEVER, IS NOT TIED TO ANY ONE OPERATING SYSTEM OR MACHINE; AND ALTHOUGH IT HAS BEEN CALLED A "SYSTEM PROGRAMMING LANGUAGE" BECAUSE IT IS USEFUL FOR WRITING OPERATING SYSTEMS, IT HAS BEEN USED EQUALLY WELL TO WRITE MAJOR NUMERICAL, TEXT-PROCESSING, AND DATA-BASE PROGRAMS.

C IS A RELATIVELY "LOW LEVEL" LANGUAGE. THIS CHARACTERIZATION IS NOT PEJORATIVE; IT SIMPLY MEANS THAT C DEALS WITH THE SAME SORT OF OBJECTS THAT MOST COMPUTERS DO, NAMELY CHARACTERS, NUMBERS, AND ADDRESSES. THESE MAY BE COMBINED AND MOVED ABOUT WITH THE USUAL ARITHMETIC AND LOGICAL OPERATORS IMPLEMENTED BY ACTUAL MACHINES.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

THE CAMBRIDGE RING AND UNIX
DOC. TYPE: ARTICLE
DOC. DATE: 1982
AUTHORS: COLLINSON, R. P. A.
SOURCE, PUB. BY, ETC: SOFTWRE - PRACTICE & EXPERIENCE, V 12, 1982
DOCUMENT NUMBER: 0038-0644/82/060583-12
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
THE RING IS USED FOR TERMINAL ACCESS TO THE VAX, TO TRANSMIT OUTPUT TO PERIPHERALS SUCH AS PRINTERS OR PLOTTERS AND FOR FILE TRANSFER BETWEEN MACHINES ON THE SITE. THE APPROACH WHICH HAS EVOLVED ATTEMPTS TO MAKE THE NETWORK CONNECTIONS INVISIBLE TO THE SYSTEM; FOR INSTANCE, TERMINALS ROUND THE RING SEEM TO BE CONNECTED TO STANDARD SERIAL LINES, AND PRINTERS ATTACHED TO REMOTE PROCESSORS ARE ACCESSED AS IF THEY WERE DIRECTLY CONNECTED TO THE HARDWARE. THE IMPLEMENTATION IS NOT DEVASTATINGLY NEW RESEARCH BUT A PIECE OF BREAD AND BUTTER WORK DESIGNED TO SUPPORT A LARGE NUMBER OF USERS ON A UNIX SYSTEM FOR WHICH ALL PUBLIC ACCESS IS VIA TERMINALS CONNECTED TO FRONT-END PROCESSORS ON THE RING RATHER THAN DIRECTLY TO THE VAX.

THE CHORUS DISTRIBUTED OPERATING SYSTEM: DESIGN AND IMPLEMENTATION
DOC. TYPE: PAPER
DOC. DATE: 1982
AUTHORS: GUILLEMONT, M.
SOURCE, PUB. BY, ETC: IFIP 1982, LOCAL COMPUTER NETWORKS, N.-HOLLAND
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
CHORUS IS AN ARCHITECTURE FOR DISTRIBUTED SYSTEMS. IT INCLUDES A METHOD FOR DESIGNING A DISTRIBUTED APPLICATION, A STRUCTURE FOR ITS EXECUTION AND THE (OPERATING) SYSTEM TO SUPPORT THIS EXECUTION. ONE IMPORTANT CHARACTERISTIC OF CHORUS IS THAT THE MAJOR PART OF THE SYSTEM IS BUILT WITH THE SAME ARCHITECTURE AS APPLICATIONS. IN PARTICULAR, THE EXCHANGE OF MESSAGES, WHICH IS THE FUNDAMENTAL COMMUNICATION/SYNCHRONIZATION MECHANISM, HAS BEEN EXTENDED TO THE MOST BASIC FUNCTIONS OF THE SYSTEM.

THE DESCRIPTION OF FAULT-TOLERANT SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: 1981
AUTHORS: SYRBE, M.
SOURCE, PUB. BY, ETC: PROCESS AUTOMATION (GERMANY), NO. 1
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
VARIOUS DEMANDS OF INCREASING COMPLEXITY AND THE DISPOSABILITY OF NEW TECHNOLOGIES, LIKE THE ONE-CHIP-MICROCOMPUTER AND FIBRE OPTICS, LEAD TO CONTROL SYSTEMS WHICH ARE BUILD AS DECENTRALISED, DISTRIBUTED MULTI-MICROCOMPUTERS SYSTEMS. THEY REALISE NOT ONLY NEW CONTROL FUNCTIONS BUT THEY ALSO OPEN UP POSSIBILITIES OF INCREASING AVAILABILITY BY FAULT-TOLERANCE. THE DESIGN OR THE SELECTION AND LAY-OUT OF SUCH SYSTEMS REQUIRE A QUANTITATIVE DESCRIPTION OF THESE SYSTEMS. THIS IS POSSIBLE ON THE BASIS OF THE SET OF QUEUEING MODELS, RELIABILITY NETS AND DIAGNOSTIC GRAPHS. THIS IS SHOWN BY AN EXAMPLE OF A PRACTICALLY APPLIED REALLY DISTRIBUTED COMPUTER CONTROL SYSTEM (RDC-SYSTEM). COMPUTER-AIDED METHODS FOR THESE SYSTEM DESCRIPTIONS ARE EMPHASISED.

PROF. DR. M. SYRBE IS WITH FRAUNHOFER-INSTITUT FUR INFORMATIONS- UND DATENVERARBEITUNG, 11TB, SEBASTIAN-KNEIPP-STRASSE 12/14, D-7500, KARLSRUHE1.

THE EFFECTS OF ADA AND THE INTEL IAPX432 UPON THE MANAGEMENT OF COMPUTER RESOURCES
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1983
AUTHORS: HART, D. A.
SOURCE, PUB. BY, ETC: AIAA COMPUTERS IN AEROSPACE IV CONFERENCE
DOCUMENT NUMBER: AIAA-83-2340-CP (NOT INCL IN CONF PROC'GS)
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE MANAGEMENT OF COMPUTING RESOURCES IS SHOWN TO BE MOVING INTO AN IMPORTANT NEW PHASE WITH THE INTRODUCTION OF THE ADA LANGUAGE AND ITS PROGRAMMING SUPPORT ENVIRONMENT TOGETHER WITH THE INTEL IAPX432 MICROCOMPUTER. THE STRENGTH OF SOFTWARE ENGINEERING AS A FOUNDATION FOR STRUCTURING SOFTWARE DEVELOPMENT ENVIRONMENTS WITH HIGHER LEVELS OF PRODUCTIVITY AND QUALITY IS IDENTIFIED AS A PRIMARY VERIABLE IN THIS MOVEMENT.

THE AUTHOR IS WITH THE UNIVERSITY OF HOUSTON - CLEAR LAKE, HOUSTON, TX.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

THE EXECUTION UNIT FOR THE VLSI 432 GENERAL DATA PROCESSOR
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: BUDDE, D. L. COLLEY, S. R. DOMENIK, S. L. ET AL
SOURCE, PUB. BY, ETC: IEEE J. OF SOLID-STATE CIRCUITS, V SC-L6, #5
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE MICROINSTRUCTION EXECUTION UNIT (MEU), WHICH IS ONE OF TWO CHIPS THAT IMPLEMENT A 32-BIT VLSI OBJECT-ORIENTED GENERAL DATA PROCESSOR, IS DESCRIBED. A COMPANION PAPER DESCRIBES THE INSTRUCTION DECODING UNIT (IDU) THAT PROVIDES THE MICROINSTRUCTION INPUTS TO BE EXECUTED BY THE MEU. A FUNCTIONAL OVERVIEW OF THE MEU IS GIVEN AND IMPORTANT DESIGN TRADEOFFS ARE DESCRIBED, INCLUDING SOME OF THE MOTIVATIONS FOR THE PARTICULAR PARTITIONING OF THE FUNCTIONS BETWEEN THE TWO CHIPS. FINALLY, DETAILS OF CIRCUIT IMPLEMENTATIONS ARE DESCRIBED FOR SOME OF THE MORE INTERESTING AND IMPORTANT CIRCUITS ON THE CHIP.

AUTHORS ARE: BUDDE, D. L.; COLLEY, S. R.; DOMENIK, S. L.; GOODMAN, A. L.; HOWARD, J. D.; AND IMEL, M. T.

THE IAPX432, A NEXT GENERATION MICROPROCESSOR
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: VAN RUMSTE, M.
SOURCE, PUB. BY, ETC: MICROPROCESSING AND MICROPROGRAMMING 11 (1983)
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THIS PAPER INTENDS TO INTRODUCE THE FUNDAMENTAL PRINCIPLES BEHIND THE ARCHITECTURE OF THE INTEL IAPX432 MICROMAINFRAME. AFTER A SHORT HISTORICAL OVERVIEW, DESCRIBING TYPICAL PROBLEMS IN EXISTING COMPUTER SYSTEMS, WE INTRODUCE THE OBJECT ORIENTED ARCHITECTURE OF THEIAPX432. THIS ARCHITECTURE PROVIDES MECHANISMS FOR AN EFFICIENT IMPLEMENTATION OF HIGH LEVEL PROGRAMMING LANGUAGES, PROCESS PARALLELISM OF CONCURRENCY, PROCESS COMMUNICATION AND TRANSPARENT MULTIPROCESSOR SYSTEMS. AFTER THIS INTRODUCTION, WE DISCUSS BRIEFLY THE PROGRAMMING LANGUAGE ADA, AND THE CORRESPONDENCE BETWEEN SOME ADA CONSTRUCTS AND 432 OBJECTS.

THE PUBLISHER IS NORTH-HOLLAND PUBLISHING CO. THIS PAPER WAS DELIVERED AS THE KEYNOTE ADDRESS AT THE 8TH SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING, 9-10 DECEMBER 1982, ANTWERP, BELGIUM.

THE AUTHOR IS WITH CENTRAL LAB, BELL TELEPHONE MFG. CO., FRANCIS WELLESPLEIN 1, B-2000 ANTWERP, BELGIUM.

THE INTEGRATION OF USER PERCEPTION IN THE HETEROGENEOUS M/M/2 QUEUE
DOC. TYPE: PAPER
AUTHORS: GEIST, R. TRIVEDI, K.
SOURCE, PUB. BY, ETC: IFIP WG 7.3 INT SYM ON PERFORMANCE MODELING
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
WE CONSIDER THE USE OF A NEWLY PROPOSED MEASURE OF SYSTEM CONGESTION, THE PERCEIVED MEAN, AS THE OBJECTIVE FUNCTION IN A COLLECTION OF CONFIGURATION DESIGN PROBLEMS BASED UPON THE HETEROGENEOUS M/M/2 QUEUE. WE FIND THAT THE RESULTING SOLUTIONS ARE UNIFORMLY EASIER TO DESCRIBE AND IMPLEMENT THAN THOSE OBTAINED THROUGH A CLASSICAL ANALYSIS. THIS APPROACH ALSO LEADS TO SOME INTERESTING SUGGESTIONS AS TO HOW SYSTEM USERS ACTUALLY PERCIEVE SYSTEM PERFORMANCE.

THE IUS PAPERS
DOC. TYPE: REPORT
DOC. DATE: OCTOBER 1981
AUTHORS: DALY, K. C. HARRISON, J. V. A. GAI, E. G.
SOURCE, PUB. BY, ETC: CSDL
DOCUMENT NUMBER: R-1516
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
THIS REPORT BRINGS TOGETHER THE FIVE PAPERS THAT WERE GENERATED DURING THE 3 YEAR PARTICIPATION OF CSDL IN THE DESIGN OF THE REDUNDANT NAVIGATION SYSTEM FOR THE INERTIAL UPPER STAGE (IUS).

IN THE FIFTH PAPER A MARKOV MODEL IS DEVELOPED TO OBTAIN AN OVERALL FIGURE-OF-MERIT FOR BOTH ACCURACY AND RELIABILITY, USING THE RESULTS THAT WERE DEVELOPED IN THE PREVIOUS FOUR PAPERS. THIS MODEL IS USED TO SELECT SUITABLE FDI THRESHOLDS FOR THE SYSTEM AND TO EVALUATE THE MARGINS INHERENT IN THE DESIGN GIVEN A PARTICULAR CHOICE OF VALUES FOR THE DESIGN PARAMETERS.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

THE MEASUREMENT AND ANALYSIS OF TRANSIENT ERRORS IN DIGITAL COMPUTER SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: JUNE 1979
AUTHORS: MCCONNEL, S. R. SIEWIOREK, DANIEL P.
SOURCE, PUB. BY, ETC: 9TH ANN INT SYM ON FAULT-TOLERANT COMPUTING
DOCUMENT NUMBER: CH1396-1/79/0000-0067
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
EXPERIMENTAL DATA ON TRANSIENT FAULTS FROM SEVERAL DIGITAL COMPUTER SYSTEMS ARE PRESENTED AND ANALYZED. THIS RESEARCH IS SIGNIFICANT BECAUSE EARLIER WORK ON VALIDATION OF RELIABILITY MODELS HAS CONCENTRATED ONLY ON PERMANENT FAULTS. THE SYSTEMS FOR WHICH DATA HAVE BEEN COLLECTED ARE THE DEC PDP-10 SERIES COMPUTERS, THE C.VMP FAULT TOLERANT MICROPROCESSOR, AND THE CM MULTIPROCESSOR. CURRENT RESULTS SHOW THAT TRANSIENT FAULTS DO NOT OCCUR WITH CONSTANT FAILURE RATES AS HAS BEEN COMMONLY ASSUMED. INSTEAD, THE DATA FOR ALL THREE SYSTEMS INDICATE WEIBULL DISTRIBUTIONS WITH DECREASING FAILURE RATES.

THE MISSING LINK FOR ADVANCED AVIONICS SYSTEMS EXECUTIVES
DOC. TYPE: PAPER
DOC. DATE: 1983
AUTHORS: LEEPER, K.R.
SOURCE, PUB. BY, ETC: PROC IEEE/AIAA 5TH DIG AV SYS CONF, 10/31-11/03/83
FILED BY (NAME): WERNER, R.
ABSTRACT:
THE BOEING MILITARY AIRPLANE COMPANY IS INVOLVED IN AN INDEPENDENT RESEARCH AND DEVELOPMENT EFFORT TO PROVIDE AN ADVANCED CAPABILITY IN EXECUTIVE SOFTWARE. THE GOAL IS TO DEVELOP A RELIABLE AND FLEXIBLE AVIONICS EXECUTIVE COMPATIBLE WITH EXISTING AND ANTICIPATED ADVANCED AVIONICS ARCHITECTURES. AS ONE LINK IN THE CHAIN OF EVOLUTIONARY MODIFICATIONS LEADING TO INCREASINGLY ADVANCED CAPABILITY, BMAC HAS REHOSTED THE SINGLE PROCESSOR SYNCHRONOUS EXECUTIVE TO OPERATE ON A MIL-STD-1750A PROCESSOR. THIS PAPER DESCRIBES THE HISTORY OF THE SPSE, DISCUSSES THE MODIFICATION PROCESS AND RESULTS, AND SUGGESTS THE DIRECTION IN WHICH FUTURE EVOLUTIONARY CHANGES ARE EXPECTED TO GO.

THE CONFERENCE AT WHICH THIS PAPER WAS PRESENTED WAS HELD IN SEATTLE, WASHINGTON.

THE MYTHICAL MAN-MONTH
DOC. TYPE: BOOKS
DOC. DATE: JULY 1, 1978
AUTHORS: BROOKS, F.P.
SOURCE, PUB. BY, ETC: ADDISON-WESLEY PUBLISHING COMPANY
DOCUMENT NUMBER: ISBN 0-201-00650-2
FILED BY (NAME): KNOSP, A.

THE NEWCASTLE CONNECTION OR UNIXES OF THE WORLD UNITE!
DOC. TYPE: ARTICLE
DOC. DATE: DECEMBER 1982
AUTHORS: BROWNBRIDGE, D. MARSHALL, L. RANDELL, B.
SOURCE, PUB. BY, ETC: SOFTWARE-PRAC. & EXPERIENCE; 12, P 1147 '82
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
THE PAPER DESCRIBES A SOFTWARE SUBSYSTEM THAT CAN BE ADDED TO EACH OF A SET OF PHYSICALLY INTERCONNECTED UNIX OR UNIX LOOK-ALIKE SYSTEMS, SO AS TO CONSTRUCT A DISTRIBUTED SYSTEM WHICH IS FUNCTIONALLY INDISTINGUISHABLE AT BOTH THE USER AND THE PROGRAM LEVEL FROM A CONVENTIONAL SINGLE-PROCESSOR UNIX SYSTEM. THE TECHNIQUES USED ARE APPLICABLE TO A VARIETY AND MULTIPLICITY OF BOTH LOCAL AND WIDE-AREA NETWORKS, AND ENABLE ALL ISSUES OF INTER-PROCESSOR COMMUNICATION, NETWORK PROTOCOLS, ETC., TO BE HIDDEN. A BRIEF ACCOUNT IS GIVEN OF EXPERIENCE WITH SUCH A DESTRIBUTED SYSTEM, WHICH IS CURRENTLY OPERATIONAL ON A SET OF PDP11S CONNECTED BY A CAMBRIDGE RING. THE FINAL SECTIONS COMPARE THE SCHEME DESCRIBED TO VARIOUS PRECURSOR SCHEMES AND DISCUSS ITS POTENTIAL RELEVANCE TO OTHER OPERATING SYSTEMS. THE WORK HAS MADE CLEAR THAT THE STRUCTURE AND MECHANISMS OF A MULTIPROCESSING OPERATING SYSTEM ARE VERY SIMILAR TO THOSE OF DISTRIBUTED SYSTEMS, JUST AS THEY HAVE BEEN KNOWN TO BE VERY SIMILAR TO THOSE OF A (GOOD) MULTIPROGRAMMING SYSTEM. (SOURCE: SOFTWARE - PRACTICE AND EXPERIENCE, JOHN WILEY & SONS, LTD.)

THE RACE BETWEEN C AND ADA MAY HAVE TWO WINNERS
DOC. TYPE: ARTICLE
DOC. DATE: APRIL 14, 1983
AUTHORS: MUELLER, FREDERICK R. TAFT, S. TUCKER
SOURCE, PUB. BY, ETC: ELECTRONIC DESIGN 4/14/83, HAYDEN PUBLISHING CO.
FILED BY (NAME): KNOSP, A. A.
ABSTRACT:
C AND ADA, BOTH DESIGNED FOR SYSTEM SOFTWARE, SHARE MANY CONCEPTS BUT EMBODY OPPOSITE PHILOSOPHIES. EXAMPLES DEMONSTRATE WHICH DOES A BETTER JOB IN WHAT APPLICATION. C TRADES SAFETY FOR SIMPLICITY; IT IS EXTREMELY FLEXIBLE, BUT REQUIRES THE USE OF EXTERNAL SUBROUTINE FOR I/O AND SYSTEMS FUNCTIONS. ADA ATTEMPTS TO BE COMPREHENSIVE BY INCLUDING REAL-TIME TASKING CONSTRUCTS, HIGH AND LOW LEVEL I/O CONSTRUCTS, STRONG TYPING AND INTERFACE SPECIFICATION AND CHECKING.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

THE THEORY AND PRACTICE OF RELIABLE SYSTEM DESIGN
DOC. TYPE: BOOK
DOC. DATE: 1982
AUTHORS: SIEWIOREK, DANIEL P. SWARZ, ROBERT S.
SOURCE, PUB. BY, ETC: DIGITAL EQUIPMENT CORPORATION (DIGITAL PRESS)
FILED BY (NAME): LALA, J. H.
ABSTRACT:
THE BOOK IS DIVIDED INTO TWO PARTS. PART I DEALS WITH THE THEORY AND PART II WITH THE PRACTICE OF RELIABLE DESIGN. THE APPENDICES PROVIDE DETAILED INFORMATION ON CODING THEORY, DESIGN FOR TESTABILITY, AND MIL-HDBK-217 COMPONENT RELIABILITY MODEL.

THE BOOK HAS THREE AUDIENCES: 1) THE FIRST IS THE ADVANCED UNDERGRADUATE INTERESTED IN RELIABLE DESIGN; 2) THE SECOND IS THE GRADUATE STUDENT SEEKING A SECOND COURSE IN RELIABLE DESIGN, PERHAPS AS A PRELUDE TO DOING RESEARCH; 3) THE THIRD IS THE PRACTICING ENGINEER. A MAJOR GOAL OF THE BOOK IS TO PROVIDE ENOUGH CONCEPTS TO ENABLE THE PRACTICING ENGINEER TO INCORPORATE COMPREHENSIVE RELIABILITY TECHNIQUES INTO HIS OR HER NEXT DESIGN.

THE FINAL CHAPTER OF THE BOOK PROVIDES A METHODOLOGY FOR RELIABLE SYSTEM DESIGN AND ILLUSTRATES HOW THIS METHODOLOGY CAN BE APPLIED IN AN ACTUAL DESIGN SITUATION (THE INTEL 432).

THE TRAFFIC FLOW IN A DISTRIBUTED REALTIME COMPUTING SYSTEM (RDC-SYSTEM) WITH A FIBEROPTIC RINGBUS SYSTEM
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1981
AUTHORS: HEGER, DIRK BAHRE, REINHARD
SOURCE, PUB. BY, ETC: AGARD CONF PROC #303; NORWAY, 22-25 JUNE 1981
DOCUMENT NUMBER: AGARD-CP-303; ISBN 92-835-0302-3
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
TIC #AD-A109-274. THE AUTHORS ARE CITIZENS OF GERMANY.

THE UNIX TIME-SHARING SYSTEM
DOC. TYPE: PAPER
DOC. DATE: 1974
AUTHORS: RITCHIE, D.
SOURCE, PUB. BY, ETC: COMMUNICATIONS OF THE ACM
FILED BY (NAME): FURTEK, F. C.

THE 757/767 FLIGHT MANAGEMENT SYSTEM LABORATORY TEST PROGRAM
DOC. TYPE: PAPER
AUTHORS: SPRADLIN, R.
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THE BOEING 767 AND 757 AIRPLANES, SCHEDULED TO ENTER AIRLINE SERVICE IN 1982 AND 1983 RESPECTIVELY, ARE EQUIPPED WITH AN ALL DIGITAL FLIGHT MANAGEMENT SYSTEM (FMS). THE FMS COMPRISES THE TRADITIONAL AVIONICS AND AUTOMATIC FLIGHT CONTROL FUNCTIONS, AN AREA NAVIGATION/PERFORMANCE MANAGEMENT SYSTEM AND COLOR CRT PRIMARY FLIGHT INSTRUMENTS IN THE FLIGHT DECK. THE FMS SUBSYSTEMS WERE DEVELOPED AND TESTED BY AVIONICS MANUFACTURERS IN RESPONSE TO BOEING SPECIFICATIONS; THEN TESTED AGAIN, INDIVIDUALLY AND IN COMBINATION WITH OTHER SUBSYSTEMS, BY BOEING IN ITS SEATTLE AREA LABORATORIES. THE SEATTLE TEST PROGRAM INCLUDED EXTENSIVE SIMULATOR AND LABORATORY TESTS OF EQUIPMENT HARDWARE/SOFTWARE PERFORMANCE AND SYSTEM FUNCTIONS. THE BOEING LABORATORY TEST PROGRAM IS DESCRIBED, EMPHASIZING SPECIALLY DEVELOPED TEST FACILITIES, THE TEST PLANS, A SUBSYSTEM TEST EXAMPLE AND INITIAL FLIGHT TEST RESULTS.

TIME, CLOCKS, AND THE ORDERING OF EVENTS IN A DISTRIBUTED SYSTEM
DOC. TYPE: PAPER
DOC. DATE: JULY 1978
AUTHORS: LAMPORT, L.
SOURCE, PUB. BY, ETC: COMMUNICATIONS OF THE ACM VOL 21 #7
FILED BY (NAME): FELLEMAN, P. G.
ABSTRACT:
THE CONCEPT OF ONE EVENT HAPPENING BEFORE ANOTHER IN A DISTRIBUTED SYSTEM IS EXAMINED, AND IS SHOWN TO DEFINE A PARTIAL ORDERING OF THE EVENTS. A DISTRIBUTED ALGORITHM IS GIVEN FOR SYNCHRONIZING A SYSTEM OF LOGICAL CLOCKS WHICH CAN BE USED TO TOTALLY ORDER THE EVENTS. THE USE OF THE TOTAL ORDERING IS ILLUSTRATED WITH A METHOD FOR SOLVING SYNCHRONIZATION PROBLEMS. THE ALGORITHM IS THEN SPECIALIZED FOR SYNCHRONIZING PHYSICAL CLOCKS, AND A BOUND IS DERIVED ON HOW FAR OUT OF SYNCHRONY THE CLOCKS CAN BECOME.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

TOOLS FOR SUPPORTING STRUCTURED ANALYSIS
DOC. TYPE: BOOK SECTION
DOC. DATE: 1982
AUTHORS: DELISLE, N. M. MENICOSY, D. E. KERTH, N. L.
SOURCE, PUB. BY, ETC: BOOK: AUTOMATED TOOLS FOR INFO. SYSTEMS DESIGN
FILED BY (NAME): SZULEWSKI, P.
ABSTRACT:
THIS IS A SUBJECT TREATED IN THE BOOK "AUTOMATED TOOLS FOR INFORMATION SYSTEMS DESIGN", EDITED BY H.J.SCHNEIDER AND A.I.WASSERMAN, PUBLISHED BY THE NORTH-HOLLAND PUBLISHING COMPANY.

TOTALLY DISTRIBUTED, REDUNDANTLY STRUCTURED HARDWARE AND SOFTWARE LOCAL COMPUTER CONTROL NETWORK
DOC. TYPE: PAPER
DOC. DATE: 1982
AUTHORS: DAMSKER, D.
SOURCE, PUB. BY, ETC: IFIP LOCAL COMPUTER NETWORKS, 1982, N.-HOLLAND
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
THE FOLLOWING PRESENTATION OF A NEWLY DEVISED LOCAL COMPUTER CONTROL NETWORK IS BASED ON THE CONCEPTUAL DESIGN DEVELOPED FOR A SOLAR COGENERATION INDUSTRIAL PLANT, SPONSORED BY DOE UNDER CONTRACT DE-AC03-81 SF 11533. THE CONTROL SYSTEM FEATURES A MODULAR LOCAL COMPUTER NETWORK, WHOSE REDUNDANCY CAN BE DESIGNED TO MEET ANY AVAILABILITY REQUIREMENT OF A SPECIFIC APPLICATION, USING THE SAME HARDWARE BUILDING BLOCKS. THE INTRINSIC REDUNDANCY APPLIES TO THE COMPONENT LEVEL, WITH AN ORDERLY OVERLAPPING, AS OPPOSED TO A COMPLETE SYSTEM REDUNDANCY WHICH IS MORE COSTLY AND LESS RELIABLE. THE CONTROL SOFTWARE IS HIERARCHICAL AND DISTRIBUTED ORGANIZED WITH A REDUNDANT STRUCTURE AND TOPOLOGY. THE LOCAL COMPUTER NETWORK USES SIMPLE PACKET SWITCHING AND ROUTING DECISION-MAKING TECHNIQUES, BASED ON A CONTENTION METHOD, WHICH PRESENTS THE ADVANTAGE OF CONVEYING MESSAGES IN THE ORDER OF THEIR PRIORITY. SOME DETAILS OF THE CONCEPT ARE PRESENTED AS THEY WERE APPLIED TO THE SOLAR COGENERATION CONCEPTUAL DESIGN.

TRENDS IN RELIABILITY MODELING TECHNOLOGY FOR FAULT-TOLERANT SYSTEMS
DOC. TYPE: TECH MEMO
DOC. DATE: APRIL 1979
AUTHORS: BAVUSO, S. J.
SOURCE, PUB. BY, ETC: NASA LANGLEY RESEARCH CENTER
DOCUMENT NUMBER: NASA TM 80089
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
RECENT DEVELOPMENTS IN RELIABILITY MODELING FOR FAULT TOLERANT AVIONIC COMPUTING SYSTEMS ARE PRESENTED. EMPHASIS IS PLACED ON THE MODELING OF LARGE SYSTEMS WHERE ISSUES OF STATE SIZE AND COMPLEXITY, FAULT COVERAGE, AND PRACTICAL COMPUTATION ARE ADDRESSED.

A GENERAL DISCUSSION OF FAULT COVERAGE AND HOW IT IMPACTS SYSTEM DESIGN IS PRESENTED TOGETHER WITH A HISTORICAL ACCOUNT OF THE RESEARCH WHICH LED TO THE CURRENT FAULT COVERAGE DEVELOPMENTAL PROGRAM..SEVERAL ASPECTS OF FAULT COVERAGE INCLUDING MODELING AND DATA MEASUREMENT OF INTERMITTENT/TRANSIENT FAULTS AND LATENT FAULTS ARE ELUCIDATED AND ILLUSTRATED. THE CARE II(COMPUTER-AIDED RELIABILITY ESTIMATION) COVERAGE IS PRESENTED AND SHORTCOMINGS TO BE ELIMINATED IN THE FUTURE CARE III ARE DISCUSSED.

THE EMERGENCE OF THE SO-CALLED LATENT FAULT AS A SIGNIFICANT FACTOR IN RELIABILITY ASSESSMENT IS GAINING INCREASED ATTENTION FROM A MODELING VIEWPOINT; THEREFORE, NUANCES OF LATENT FAULTS, MODELS FOR SUCH, AND A METHOD FOR LATENT FAULT MEASUREMENT ARE DEPICTED.

TRENDS IN THE DESIGN AND IMPLEMENTATION OF PROGRAMMING LANGUAGES
DOC. TYPE: ARTICLE
DOC. DATE: JANUARY 1980
AUTHORS: WULF, WILLIAM A.
SOURCE, PUB. BY, ETC: COMPUTER JAN 1980 PAGE 14-22, IEEE COMPTR SOCIETY
FILED BY (NAME): KNOSP, A. A.
ABSTRACT:
BY THE END OF THE DECADE, ADVANCES IN AREAS SUCH AS STRUCTURED PROGRAMMING, SPECIFICATION, VERIFICATION, AND LANGUAGE DESIGN WILL HAVE TRANSFORMED PROGRAMMING INTO A TRUE ENGINEERING DISCIPLINE.

TWO CHIPS ENDOW 32-BIT PROCESSOR WITH FAULT-TOLERANT ARCHITECTURE
DOC. TYPE: ARTICLE
DOC. DATE: APRIL 7, 1983
AUTHORS: PETERSON, C. DUZETTE, C. BUDDE, D. ET AL
SOURCE, PUB. BY, ETC: ELECTRONICS, 04/07/83
FILED BY (NAME): KEMP, A.
ABSTRACT:
(AUTHORS ARE: PETERSON, C.; DUZETTE, R.; BUDDE, D.; CARSON, D.; IMEL, M.; JASPER, C.; JOHNSON, D.; KRAVITZ, R.; NG, C.; WILDE, D.; AND YOUNG, J. ALL AUTHORS ARE WITH INTEL CORP.)

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

ULTRA-HIGH RELIABILITY PREDICTION FOR FAULT-TOLERANT COMPUTER SYSTEMS
DOC. TYPE: ARTICLE
DOC. DATE: FEBRUARY 1983
AUTHORS: GEIST, R. TRIVEDI, K.
SOURCE, PUB. BY, ETC: IEEE TRANS ON COMPUTERS
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
A REVIEW AND A CRITICAL EVALUATION OF A REPRESENTATIVE CLASS OF STATE-OF-THE-ART MODELS FOR ULTRA-HIGH RELIABILITY PREDICTION IS PRESENTED. THIS EVALUATION NATURALLY LEADS US TO A NEW MODEL FOR ULTRA-HIGH RELIABILITY PREDICTION NOW UNDER DEVELOPMENT. THE NEW MODEL COMBINES THE FLEXIBILITY AND ACCURACY OF SIMULATION WITH THE SPEED OF ANALYTIC MODELS.

USE OF FAULT TREES FOR THE DESIGN OF RECOVERY BLOCKS
DOC. TYPE: PAPER
DOC. DATE: JUNE 22, 1982
AUTHORS: HECHT, H. HECHT, M.
SOURCE, PUB. BY, ETC: 12TH ANN INTL SYMP ON FAULT-TOLERANT COMPUTING
DOCUMENT NUMBER: (PROCEEDINGS: TIC# QA 76.5 .I58 1982)
FILED BY (NAME): FURTEK, F. C.
ABSTRACT:
THE SCOPE OF RECOVERY BLOCKS AND THE PLACEMENT OF ACCEPTANCE TESTS HAD IN PRIOR WORK USUALLY BEEN DICTATED BY THE FUNCTIONAL REQUIREMENTS OF THE PROGRAM. IN THE DESIGN OF FAULT TOLERANT SOFTWARE FOR SYSTEM PROGRAMS (SCHEDULERS, HARDWARE RECONFIGURATION, ETC.), DIFFERENT GUIDELINES ARE REQUIRED. FAULT TREES WERE FOUND TO BE WELL SUITED. THEIR USE IS DESCRIBED, AND DIFFERENCES FROM PREVIOUS APPROACHES ARE IDENTIFIED.

VALIDATION METHODS RESEARCH FOR FAULT-TOLERANT AVIONICS AND CONTROL SYSTEM SUB-WORKING GROUP MEETING - CARE III PEER REVIEW
DOC. TYPE: REPORT
DOC. DATE: SEPTEMBER 1980
AUTHORS: TRIVEDI, K. CLARY, J.
SOURCE, PUB. BY, ETC: RESEARCH TRIANGLE PARK
DOCUMENT NUMBER: NASA CONF PUBLICATION 2167
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
ON 15-16 SEPTEMBER 1980, A SUB-WORKING GROUP MEETING WAS HELD AT THE RESEARCH TRIANGLE INSTITUTE, RESEARCH TRIANGLE PARK, NORTH CAROLINA TO:

1. CONDUCT A PEER REVIEW OF CARE III, INCLUDING AN EXAMINATION OF THE ASSUMPTIONS ON WHICH CARE III IS BASED AND THE FUNDAMENTAL PROBABILISTIC NOTIONS BEHIND IT.

2. EVALUATE CARE III'S EFFECTIVENESS; IN MEETING ITS GOALS; NAMELY, TO MODEL ACCURATELY THE BEHAVIOR OF ULTRARELIABLE SYSTEMS REQUIRED BY FLIGHT-CRITICAL AVIONICS AND CONTROL SYSTEMS.

3. RECOMMEND TESTS THAT EXPLORE AND VALIDATE THE CAPABILITIES OF CARE III.

VALIDATION, VERIFICATION, AND TESTING OF COMPUTER SOFTWARE
DOC. TYPE: ARTICLE
DOC. DATE: JUNE 2, 1982
AUTHORS: ADRION, W. BRANSTAD, M. CHERNIAVSKY, J.
SOURCE, PUB. BY, ETC: COMPUTING SURVEYS V.14 #2
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
SOFTWARE QUALITY IS ACHIEVED THROUGH THE APPLICATION OF DEVELOPMENT TECHNIQUES AND THE USE OF VERIFICATION PROCEDURES THROUGHOUT THE DEVELOPMENT PROCESS. CAREFUL CONSIDERATION OF SPECIFIC QUALITY ATTRIBUTES AND VALIDATION REQUIREMENTS LEADS TO THE SELECTION OF A BALANCED COLLECTION OF REVIEW, ANALYSIS, AND TESTING TECHNIQUES FOR USE THROUGHOUT THE LIFE CYCLE. THIS PAPER SURVEYS CURRENT VERIFICATION, VALIDATION, AND TESTING APPROACHES AND DISCUSSES THEIR STRENGTHS, WEAKNESSES, AND LIFE-CYCLE USAGE. IN CONJUNCTION WITH THESE, THE PAPER DESCRIBES AUTOMATED TOOLS USED TO IMPLEMENT VALIDATION, VERIFICATION, AND TESTING. IN THE DISCUSSION OF NEW RESEARCH THRUSTS, EMPHASIS IS GIVEN TO THE CONTINUED NEED TO DEVELOP A STRONGER THEORETICAL BASIS FOR TESTING AND THE NEED TO EMPLOY COMBINATIONS OF TOOLS AND TECHNIQUES THAT MAY VARY OVER EACH APPLICATION.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

VERIFICATION AND VALIDATION TECHNIQUES APPLIED TO THE RECONFIGURABLE BLOCK-5D SATELLITE SOFTWARE

DOC. TYPE: REPORT
AUTHORS: DINIAK, W.
SOURCE, PUB. BY, ETC: CSDL
DOCUMENT NUMBER: USAF HQ. SPACE #PAS-0674
FILED BY (NAME): WERNER, R. E.
ABSTRACT:
THE U.S. AIR FORCE DEFENSE METEOROLOGICAL SATELLITE PROGRAM (DMSP) BLOCK 5D SATELLITE USES TWO GENERAL PURPOSE DIGITAL FLIGHT COMPUTERS FOR CONTROL OF THE SATELLITE AND LAUNCH VEHICLE. DURING THE LIFE OF SIX SATELLITES, VARIOUS RECONFIGURATIONS OF THE FLIGHT SOFTWARE WERE ACCOMPLISHED TO ACHIEVE IMPROVEMENTS OR TO COMPENSATE FOR PROBLEMS. AN INDEPENDENT VERIFICATION AND VALIDATION (IV&V) PROCESS WAS CONDUCTED BY CSDL ON THAT FLIGHT SOFTWARE PRIOR TO LAUNCH AND PRIOR TO UPLINK OF MAJOR SOFTWARE CHANGES. TO ACCOMPLISH THIS, CSDL DEVELOPED AND USED A CLOSED-LOOP DIGITAL SIMULATION OF THE MISSION TO EXERCISE THE ACTUAL FLIGHT SOFTWARE IN A REALISTIC ENVIRONMENT. THIS PAPER DESCRIBES THIS EXPERIENCE, FOCUSING ON HOW BOTH THE SIMULATION AND THE IV&V PROCESS EVOLVED AS A CONSEQUENCE OF THE FLIGHT HISTORY.

WAVELENGTH DIVISION MULTIPLEXING FOR FUTURE SPACE STATION DATA SYSTEMS

DOC. TYPE: PAPER
DOC. DATE: AUGUST 25, 1983
AUTHORS: HENDRICKS, H. MURRAY, N.
SOURCE, PUB. BY, ETC: NASA LARC; SPIE, SAN DIEGO, 08/25/83
FILED BY (NAME): KEMP, A.
ABSTRACT:
A FUTURE SPACE STATION WILL REQUIRE INFORMATION NETWORK ARCHITECTURES AND TECHNOLOGIES THAT ARE EVOLVABLE, ADAPTIVE, HIGH PERFORMING, AND SELF-CORRECTING AND -REPAIRING. ONE OF THE CONCEPTUAL NETWORK CONFIGURATIONS INVOLVES FIBER OPTIC DATA BUSES WITH WAVELENGTH DIVISION MULTIPLEXING (WDM). DISCUSSION OF THIS NETWORK CONCEPT, COMPONENTS BEING DEVELOPED AND RESULTS ON A FOUR-CHANNEL WDM STAR BUS ARE PRESENTED.

WHAT IS A "DISTRIBUTED" DATA PROCESSING SYSTEM?

DOC. TYPE: PAPER
DOC. DATE: JANUARY 1978
AUTHORS: ENSLOW, PHILIP H., JR.
FILED BY (NAME): FELLEMAN, P. G.
ABSTRACT:
AT LEAST 4 PHYSICAL COMPONENTS OF A SYSTEM MIGHT BE DISTRIBUTED: HARDWARE OR PROCESSING LOGIC, DATA, THE PROCESSING ITSELF, AND THE CONTROL (THE OPERATING SYSTEM). A PROPER DEFINITION MUST COVER THE CONCEPTS UNDER WHICH THE DISTRIBUTED COMPONENTS INTERACT. WHAT IS PRESENTED IN THIS PAPER MIGHT BE CONSIDERED THE "RESEARCH AND DEVELOPMENT" DEFINITION OF DISTRIBUTED DATA PROCESSING SYSTEMS, AND HAS 5 COMPONENTS: 1) A MULTIPLICITY OF GENERAL-PURPOSE RESOURCE COMPONENTS, INCLUDING BOTH PHYSICAL AND LOGICAL RESOURCES, THAT CAN BE ASSIGNED TO SPECIFIC TASKS ON A DYNAMIC BASIS. 2)A PHYSICAL DISTRIBUTION OF THESE PHYSICAL AND LOGICAL COMPONENTS OF THE SYSTEM INTERACTING THROUGH A COMMUNICATION NETWORK. 3)A HIGH-LEVEL OPERATING SYSTEM THAT UNIFIES AND INTEGRATES THE CONTROL OF THE DISTRIBUTED COMPONENTS. 4)SYSTEM TRANSPARENCY, PERMITTING SERVICES TO BE REQUESTED BY NAME ONLY. 5)COOPERATIVE AUTONOMY, CHARACTERIZING THE OPERATION ANDINTERACTION OF BOTH PHYSICAL AND LOGICAL RESOURCES. THESE PROPERTIES AND OPERATING CHARACTERISTICS ARE PRESENT IN A NUMBER OF SYSTEMS TO VARYING DEGREES. HOWEVER, ONLY THE COMBINATION OF ALL OF THE CRITERIA UNIQUELY DEFINES DISTRIBUTED DATA PROCESSING SYSTEMS.

WHY PASCAL IS NOT MY FAVORITE PROGRAMMING LANGUAGE

DOC. TYPE: REPORT
DOC. DATE: JULY 18, 1981
AUTHORS: KERNIGHAN, BRIAN W.
SOURCE, PUB. BY, ETC: BELL LABORATORIES, COMPUTING SCIENCE
DOCUMENT NUMBER: TECHNICAL REPORT #100
FILED BY (NAME): KNOSP, A. A.
ABSTRACT:
THE PROGRAMMING LANGUAGE PASCAL HAS BECOME THE DOMINANT LANGUAGE OF INSTRUCTION IN COMPUTER SCIENCE EDUCATION. IT HAS ALSO STRONGLY INFLUENCED LANGUAGES DEVELOPED SUBSEQUENTLY, IN PARTICULAR ADA.

PASCAL WAS ORIGINALLY INTENDED PRIMARILY AS A TEACHING LANGUAGE, BUT IT HAS BEEN MORE AND MORE OFTEN RECOMMENDED AS A LANGUAGE FOR SERIOUS PROGRAMMING AS WELL, FOR EXAMPLE, FOR SYSTEM PROGRAMMING TASKS AND EVEN OPERATING SYSTEMS.

PASCAL, AT LEAST IN ITS STANDARD FORM, IS JUST NOT SUITABLE FOR SERIOUS PROGRAMMING. THIS PAPER DISCUSSES MY PERSONAL DISCOVERY OF SOME OF THE REASONS WHY.

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

WORKLOAD CHARACTERIZATION AND SELECTION IN COMPUTER PERFORMANCE MEASUREMENT
DOC. TYPE: PAPER
DOC. DATE: AUGUST 7, 1972
AUTHORS: FERRARI, D.
SOURCE, PUB. BY, ETC: COMPUTER
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
WORKLOAD CHARACTERIZATION IS A CRUCIAL PROBLEM IN PERFORMANCE EVALUATION, BUT WE ARE STILL FAR FROM A SATISFACTORY SOLUTION {7}. THIS PAPER WILL REVIEW THE SOLUTIONS PROPOSED SO FAR IN THE AREA OF PERFORMANCE MEASUREMENT.

WORKLOAD, PERFORMANCE AND RELIABILITY OF DIGITAL COMPUTING SYSTEMS
DOC. TYPE: PAPER
DOC. DATE: JUNE 1981
AUTHORS: CASTILLO, X. SIEWIOREK, DANIEL P.
SOURCE, PUB. BY, ETC: 11TH ANN INTL SYMP ON FAULT-TOLERANT COMPUTING
DOCUMENT NUMBER: CH 1600-6/8/0000/0084
FILED BY (NAME): MOTYKA, P. R.
ABSTRACT:
IN THIS PAPER A NEW MODELING METHODOLOGY TO CHARACTERIZE FAILURE PROCESSES IN TIME-SHARING SYSTS DUE TO HARDWARE TRANSIENTS AND SOFTWARE ERRORS IS SUMMARIZED. THE BASIC ASSUMP MADE IS THAT THE INSTANTANEOUS FAIL RATE OF A SYS RESOURCE CAN BE APPROXIMATED BY A DETERMINISTIC FUNC OF TIME PLUS A O-MEAN STATIONARY GAUSSIAN PROCESS, BOTH DEPENDING ON THE USAGE OF THE RESOURCE CONSIDERED. THE PROBABILITY DEN FUNC OF THE TIME TO FAILURE OBTAINED UNDER THIS ASSUMP HAS A DECREASING HAZARD FUNC, PARTIALLY EXPLAINING WHY OTHER DECREASING HAZARD FUNC DENSITIES SUCH AS THE WEIBULL FIT EXPERIMENTAL DATA SO WELL. FURTHERMORE, BY CONSIDERING THE KERNEL OF THE OPERATING SYS AS A SYS RESOURCE, THIS METHODOLOGY SETS THE BASIS FOR INDEPENDENT METHODS OF EVAL THE CONTRIBUTION OF SOFTWARE TO SYS UNRELIABILITY, AND GIVES SOME NON OBVIOUS HINTS ABOUT HOW SYS RELIABILITY COULD BE IMPROVED. A REAL SYS HAS BEEN CHARACTERIZED ACCORDING TO THIS METHODOLOGY, AND AN EXTREMELY GOOD FIT BETWEEN PREDICTED AND OBSERVED BEHAVIOR HAS BEEN FOUND. ALSO, THE PREDICTED SYS BEHAVIOR ACCORDING TO THIS METHODOLOGY IS COMPARED WITH THE PREDICTIONS OF OTHER MODELS SUCH AS THE EXPONENTIAL, WEIBULL, AND PERIODIC FAILURE RATE.

---

13 COMPANIES CLOSE IN ON LAN STANDARD PROPOSAL
DOC. TYPE: ARTICLE
DOC. DATE: MARCH 1983
AUTHORS: SEHR, R.
DOCUMENT NUMBER: J K03301
FILED BY (NAME): KERNAN, J. E.
ABSTRACT:
REPORTS OF THE DEMISE OF ETHERNET MAY HAVE BEEN, LIKE MARK TWAIN'S FIRST OBITUARY, A BIT PREMATURE.

WITH SOME FINE TUNING AND BODYWORK, THE LOCAL-AREA NETWORK PROPOSED BY THE DIGITAL EQUIPMENT CORP./INTEL CORP./XEROX CORP. TROIKA NEARLY THREE YEARS AGO AND LEFT FOR DEAD BY SOME CRITICS LATE IN 1981, HAS RECEIVED A NEW BOOST WITH THE FOUNDATION OF THE IEEE 802.3 COMMITTEE WORKING DRAFT. THE DRAFT RECEIVED THE ENTHUSIASTIC ENDORSEMENT OF 13 KEY HARDWARE AND SOFTWARE VENDORS LATE LAST YEAR.

THE DRAFT IS BEING EVALUATED BY THE PARENT IEEE COMMITTEE FOR FINAL APPROVAL. BARRING ANY FURTHER CHANGES, IT WILL RESEMBLE THE ORIGINAL ETHERNET IN MANY WAYS. "THE WORKING DRAFT IS AN ENHANCEMENT OF AN ALREADY-GOOD IDEA," SAYS DON LOUGHRY, PROJECT MANAGER FOR HEWLETT-PACKARD CO.'S DATA COMMUNICATIONS DIVISION AND CHAIRMAN OF THE WORKING IEEE GROUP, REFERRING TO THE ORIGINAL ETHERNET. "WE (THE ETHERNET AND IEEE 802 COMMITTEE) WERE LIKE TWO BOATS IN THE SAME CHANNEL WITH THE SAME SENSE OF MISSION, AND NEITHER OF US KNEW WHAT THE OTHER WAS DOING," LOUGHRY SAYS.

16-BIT BIPOLAR MICROPROCESSOR MARCHES TO STANDARD INSTRUCTION SET
DOC. TYPE: ARTICLE
DOC. DATE: APRIL 7, 1983
AUTHORS: MOR, S. HINGARH, H. VORA, M. ET AL
SOURCE, PUB. BY, ETC: ELECTRONICS, 04/07/83
FILED BY (NAME): KEMP, A.
ABSTRACT:
RATIONALE FOR FLOATING-POINT IMPLEMENTATION ON 1750A CHIP. (AUTHORS ARE: MOR, S.; HINGARH, H.; VORA, M.; WILNAI, D.; MAXWELL, D.; AND LONGO, T.)

AIPS TECHNOLOGY SURVEY LITERATURE LIST SORTED BY TITLE

32-BIT 'MEGAMICRO' EXPLOITS HARDWARE VIRTUAL MEMORY AND 'RAM DISK'
DOC. TYPE: PAPER
DOC. DATE: OCTOBER 1983
AUTHORS: METCALF, S. M. FARBER, R. M.
SOURCE, PUB. BY, ETC: MINI-MICRO SYSTEMS (JOURNAL), OCTOBER 1983
FILED BY (NAME): ODONNELL, R. N.
ABSTRACT:
SIXTEEN-BIT CPU'S MAY BEJUST A STOPOVER ON THE WAY FROM 8-BIT CPUS TO THE NEXT GENERATION OF MICROCOMPUTER: 32-BIT MACHINES USING POWERFUL PROCESSORSSUCH AS NATIONAL SEMICONDUCTOR CORP, NS16032, THE INTEL CORP. IAPX286, THE MOTOROLA INC. 68010, THE BELL LABORATORIES BELLMAC 32 AND THE HEWLETT-PACKARD CO. HP-32 (MMS, AUGUST, PAGE 187). THE VALUE OF 32-BIT MACHINES IS IN THEIR MULTIUSER, MULTIASKING POWER, PROCESSING SPEED, ABILITY TO USE MAINFRAME TECHNIQUES SUCH AS HARDWARE VIRTUAL MEMORY AND ACCESS TO SOPHISTICATED SOFTWARE PREVIOUSLY AVAILABLE ONLY ON MINICOMPUTERS AND MAINFRAMES. HOW A 32-BIT MICROCOMPUTER CAN ACCOMPLISH THESE OBJECTIVES CAN BE UNDERSTOOD BY EXAMINING OVERALL ARCHITECTURE, VIRTUAL MEMORY AND "RAM DISK" IMPLEMENTATION AND THE HARDWARE/SOFTWARE ADAPTABILITY OF THE FIRST PRODUCTION MICROCOMPUTERTO INCORPORATE THE 16-32/BIT NS16032.

THE AUTHORS ARE WITH LOGICAL MICROCOMPUTER COMPANY, CHICAGO.

APPENDIX C

EXCERPTS FROM 14 APRIL 1983 MEETING AT LARC

This appendix consists of reduced-scale images of the subset of viewgraphs pertaining to the Technology Survey, excerpted from the set of viewgraphs used by the CSDL at the Preliminary Requirements Review (PRR) meeting on April 14th, 1983, at the NASA Langley Research Center.

TECHNOLOGY SURVEY

- TECHNOLOGY SURVEY TOPICS
  - ARCHITECTURE - T.B. SMITH
  - SOFTWARE - J.B. DEWOLF
  - HARDWARE - E.C. HALL
  - SYSTEMS ANALYSIS METHODOLOGIES - E.G. GAI
- THE SURVEY PRESENTATION WAS DIVIDED INTO THE ABOVE TOPICS FOR CONVENIENCE OF PRESENTATION. THE TOPICS ARE, IN FACT, INTERRELATED.

ARCHITECTURE TECHNOLOGY MATRIX

- PROCESSING ARCHITECTURE
- NETWORKING ARCHITECTURE
- COMPUTATIONAL FAULT CONTROL AND RECOVERY
- COMMUNICATION FAULT CONTROL AND RECOVERY

ISSUES

- FAULT-FREE PERFORMANCE AND USABILITY
- TRANSPARENCY AND TRACTABILITY OF FAULT HANDLING

PROCESSING ARCHITECTURE

- SYSTEM ARCHITECTURE
  - CENTRALLY ORIENTED
  - DISTRIBUTED (MULTI-COMPUTER)
- PROCESSOR ARCHITECTURE
  - MULTIPROCESSOR
  - MONOPROCESSOR
- ISA ARCHITECTURE
  - SINGLE-LANGUAGE MACHINE (ADA, HAL/S)
  - HOL ENGINES
  - SOFTWARE/HARDWARE STANDARD

NETWORKING ARCHITECTURE

- TOPOLOGIES
  - BUS · RING · MESH · STAR
- OPERATIONAL ALTERNATIVES
  - PACKET OR CIRCUIT SWITCHING
  - TIME-DIVISION MULTIPLEXING
  - FREQUENCY/WAVE-DIVISION MULTIPLEXING
  - SPACE-DIVISION MULTIPLEXING
- SOFTWARE INTERFACE
  - SINGLE DISTRIBUTED OPERATING SYSTEM
  - MULTIPLE COMMUNICATING OPERATING SYSTEMS

COMPUTATIONAL FAULT CONTROL AND RECOVERY

- TEMPORAL CHECKS (SELF-CHECK PROGRAMMING, RETRY)
- INTEGRAL CHECKS (SELF-CHECKING LOGIC, CODES, ETC.)
- REPLICATION, COMPARISON AND VOTING
  - APPROXIMATE REPLICATION
  - EXACT OR CONGRUENT REPLICATION

ISSUES

- DEPARTURE FROM DESIRED ARCHITECTURE DUE TO FAULT HANDLING
  - PERFORMANCE
  - TRACTABILITY/ADAPTABILITY OF SOFTWARE STRUCTURE
- COMPLEXITY OF FAULT HANDLING

COMMUNICATION FAULT CONTROL AND RECOVERY

- SPATIAL REDUNDANCY (REDUNDANT DATA PATHS)
- TEMPORAL REDUNDANCY (COMMON MEDIUM REDUNDANT INFORMATION SEPARATED IN TIME, E.G. CODED REDUNDANCY)
- SPECTRAL REDUNDANCY

ISSUES

- FAULT-FREE PERFORMANCE AND USABILITY
- TIMING UPSETS DUE TO FAULT HANDLING
- DEGREE OF TRANSPARENCY

SOFTWARE TECHNOLOGY TOPICS

- FAULT-TOLERANT SOFTWARE
- DISTRIBUTED OPERATING SYSTEMS
- SOFTWARE TOOLS AND METHODS
- HIGH-ORDER LANGUAGES

FAULT-TOLERANT SOFTWARE

- SUBTOPICS:
  - N-VERSION
  - RECOVERY BLOCKS
  - OTHER APPROACHES
- EVALUATION CRITERIA:
  - EFFECTIVENESS
  - APPLICATIONS EXPERIENCE

DISTRIBUTED OPERATING SYSTEMS

- SUBTOPICS:
  - VISIBLE NETWORK
  - TRANSPARENT NETWORK
  - LAYERS/PROTOCOLS
  - DATA MANAGEMENT

- EVALUATION CRITERIA:
  - GROWTH AND CHANGE TOLERANCE
  - EFFICIENCY

SOFTWARE TOOLS AND METHODS

- SUBTOPICS:
  - REQUIREMENTS AND DESIGN TOOLS AND METHODS
  - V&V TOOLS AND METHODS
  - MANAGEMENT TOOLS AND METHODS
  - INTEGRATED ENVIRONMENTS

- EVALUATION CRITERIA:
  - RELIABILITY ENHANCEMENT
  - PRODUCTIVITY ENHANCEMENT

HIGH-ORDER LANGUAGES

- Subtopics:
  - Ada
  - HAL/S
  - Data-flow/functional languages
- Evaluation criteria
  - Availability and support
  - Impact on reliability/growth
  - Run-time efficiency

HARDWARE TECHNOLOGY SURVEY

- Processor instruction set architecture
- Semiconductor technology
- Data-net transmission medium
- Power distribution and control
- Mass memories

INSTRUCTION SET ARCHITECTURE

- RELEVANCE - DEFINES THE SOFTWARE/HARDWARE INTERFACE
- EXAMPLES - AP101, 1750A, NEBULA, MICROPROCESSORS, DATA FLOW
- SURVEY PROCEDURES
  - PERSONAL INTERVIEWS
  - LITERATURE SURVEY
- EVALUATION CRITERIA
  - MATURITY
  - HOL CAPABILITY
  - IMPLEMENTATION TECHNOLOGY
  - PROCESSING CAPACITY/DAIS MIX
  - POPULARITY/FUTURE
  - GROWTH AND CHANGE

SEMICONDUCTOR TECHNOLOGY

RELEVANCE - AIPS HARDWARE REQUIREMENTS, PACKAGING ISSUES

EXAMPLES -

MOS	BIPOLAR
• PMOS	• I^2L
• NMOS	• LOW-POWER SCHOTTKY
• MNOS	• T^2L
• CMOS	• G_AA_S

SURVEY PROCEDURES
- PERSONAL INTERVIEWS
- LIMITED INDUSTRIAL SURVEY

EVALUATION CRITERIA
- MATURITY/QUALITY/FUTURE
- SPEED/POWER
- AIPS ENVIRONMENTS
- VIABLE SOURCES
- MIL QUALIFIED

DATA-NETWORK TRANSMISSION MEDIUM

RELEVANCE - AIPS ARCHITECTURE AND MISSION REQUIREMENTS

EXAMPLES - FIBER OPTICS, TWISTED PAIR, COAX CABLE, POWER LINES

SURVEY PROCEDURES

- LITERATURE SURVEY
- PERSONAL INTERVIEWS
- ASSESS TYPICAL APPLICATIONS

EVALUATION CRITERIA

- AIPS ARCHITECTURE CONSTRAINTS
- ENVIRONMENTAL REQUIREMENTS
- TRANSMISSION BANDWIDTH
- MATURITY
- GROWTH AND CHANGE

POWER CONTROL AND DISTRIBUTION

RELEVANCE - FAULT TOLERANT AND AIPS ARCHITECTURE DEPENDENT

EXAMPLES - MESH, STAR

SURVEY PROCEDURES

- PERSONAL INTERVIEWS
- ASSESS TYPICAL APPLICATION

EVALUATION CRITERIA

- FAULT TOLERANCE
- SYSTEM ARCHITECTURE
- MISSION REQUIREMENTS
- GROWTH AND CHANGE

MASS MEMORIES

RELEVANCE - DATA RECORDING AND RETRIEVAL

EXAMPLES - BUBBLES, TAPE, DISC

SURVEY PROCEDURES

- LITERATURE SURVEY
- PERSONAL INTERVIEWS
- ASSESS APPLICATIONS

EVALUATION CRITERIA

- MATURITY
- SPEED/CAPACITY/VOLUME
- MISSION ENVIRONMENT

METHODOLOGIES SURVEY

- RELIABILITY, MAINTAINABILITY, AVAILABILITY (RMA) EVALUATION APPROACHES
- MEASURES OF MERIT
- HARDWARE AND SOFTWARE RMA MODELS
- VERIFICATION AND VALIDATION (V&V) TECHNIQUES

RELIABILITY, MAINTAINABILITY, AVAILABILITY EVALUATION APPROACHES

- SURVEY AND COMPARE GENERIC APPROACHES FOR EVALUATING RMA

- CANDIDATE APPROACHES
  - COMBINATORIAL ANALYSES
  - CONDITIONAL PROBABILITIES
  - MARKOV MODEL

- EVALUATION CRITERIA
  - COMPLEXITY
  - EFFICIENCY
  - COMPREHENSIVENESS

MEASURES OF MERIT

- SURVEY AND ASSESS MEASURES OF MERIT FOR EVALUATING RMA

- CANDIDATE MEASURES OF MERIT
  - MTBF
  - FAILURES/MILLION HOURS
  - PROBABILITY OF SUCCESS
  - N FAIL-OP
  - MEAN TIME TO REPAIR

- EVALUATION CRITERIA
  - QUALITATIVE EVALUATION RELATIVE TO THE REQUIREMENTS OF THE AIPS

HARDWARE AND SOFTWARE RMA MODELS

- Survey hardware and software RMA models

- Candidate Tools
  - CARE III
  - CARSRA
  - MARK1
  - Software models

- Evaluation criteria
  - Modifiability
  - Complexity
  - Comprehensiveness
  - Resource requirements
  - Output

VERIFICATION AND VALIDATION TECHNIQUES

- Survey V&V programs to define key concepts to aid in the definition of a practical approach for AIPS

- Evaluation criteria
  - Creation of test environment
  - Definition, implementation, and reporting of test program
  - Certification of test article

APPENDIX D

TECHNOLOGY SURVEY PRESENTATION, 28 SEPTEMBER 1983

This appendix presents reduced-scale images of the entire set of viewgraphs used by the CSDL at the AIPS Technology Survey Review (STR) presentation on September 28th, 1983, at LaRC. The information presented in Sections 1 through 4 of this report, and in the viewgraphs of this Appendix, are mutually supportive and highly correlated.

AIPS TECHNOLOGY SURVEY

REVIEW

28 SEPTEMBER 1983

TECHNOLOGY SURVEY

- IDENTIFY TECHNOLOGY REQUIRED FOR AIPS DEVELOPMENT
  - APPLICABILITY TO TARGET MISSIONS
  - RELEVANCE TO RELIABILITY AND GROWTH GOALS
- SURVEY
  - LITERATURE
  - CURRENT RESEARCH (INTERVIEWS/DISCUSSIONS)
    - NASA/DoD
    - INDUSTRY
    - ACADEMIA
- EVALUATE USEFULNESS/READINESS
- DOCUMENT: PHASE I REPORT
- TECHNOLOGY SURVEY WILL BE A CONTINUING EFFORT TO MAINTAIN AWARENESS OF TECHNOLOGY DEVELOPMENT

STATUS

- Formal Survey Completed: We will Continue to Follow-Up on Promising Subjects
- Results of Survey
  - A Pool of Information from which we will draw
  - Technology Choices Identified
  - Some Technology Choices Made

TECHNOLOGY SURVEY

- Technology Survey Topics
  - Architecture - J.H. Lala
  - Software - J.B. DeWolf
  - Hardware - J.F. McKenna
  - Systems Analysis Methodologies - E.G. Gai
  - Verification and Validation - R.E. Werner
- The presentation was divided into the above topics for convenience of presentation. The topics are, in fact, interrelated.

ARCHITECTURE TECHNOLOGY SURVEY

Visits

Organization	Main Contact	Subjects
NASA JSC	Ed Chevers	Distributed Systems, Networks
NASA LaRC	Nick Murray	Information Networks, Fiber Optic Systems
JPL	Dan Erickson	Self Checking Computer Modules
BMD	Bill McDonald	Distributed Systems Testbed, Multiprocessors
IBM	Bill Carter	Modern Commercial ISA
COLLINS	Ron Coffin	Modern Avionics ISA

ARCHITECTURE TECHNOLOGY SURVEY

Other Sources

- Information Exchange at Past Conferences, Symposia and Workshops
- Personal Contacts in the Fault-Tolerant Computing Community
- Extensive Library of Literature in the Area of Fault-Tolerant Computers and Avionics Systems
- Past Survey of Fault-Tolerant Systems

ARCHITECTURE TECHNOLOGY MATRIX

- Processing Architecture
- Networking Architecture
- Fault Detection, Identification, and Recovery: Computational Core
- Fault Detection, Identification, and Recovery: Communication

PROCESSING ARCHITECTURE

- System Architecture
- Processor Architecture
- ISA Architecture

SYSTEM ARCHITECTURE

- Centrally Oriented
- Distributed
- Federated

CENTRALLY ORIENTED SYSTEMS

- CSDL/Collins FTMP
- SRI/Bendix SIFT
- AFWAL/FDL CRMmFCS
- Honeywell M^2FCS

DISTRIBUTED SYSTEMS

- BMD Crossbar
- CMU Cm*, C.mmp
- JSC DSB Ring
- JPL SCCM
- AMES RAMP

FEDERATED SYSTEMS

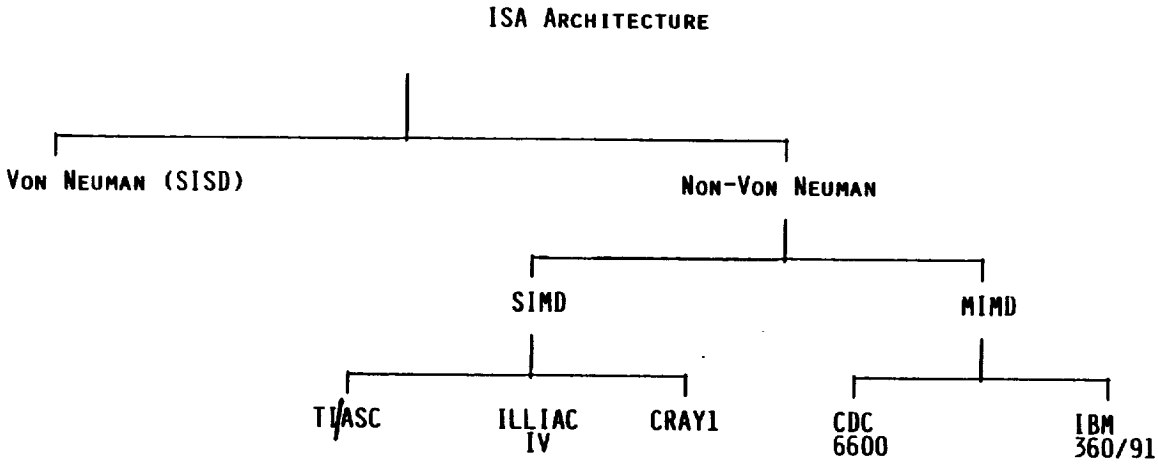
- Boeing 757/767 Avionics System
- Space Shuttle Avionics System

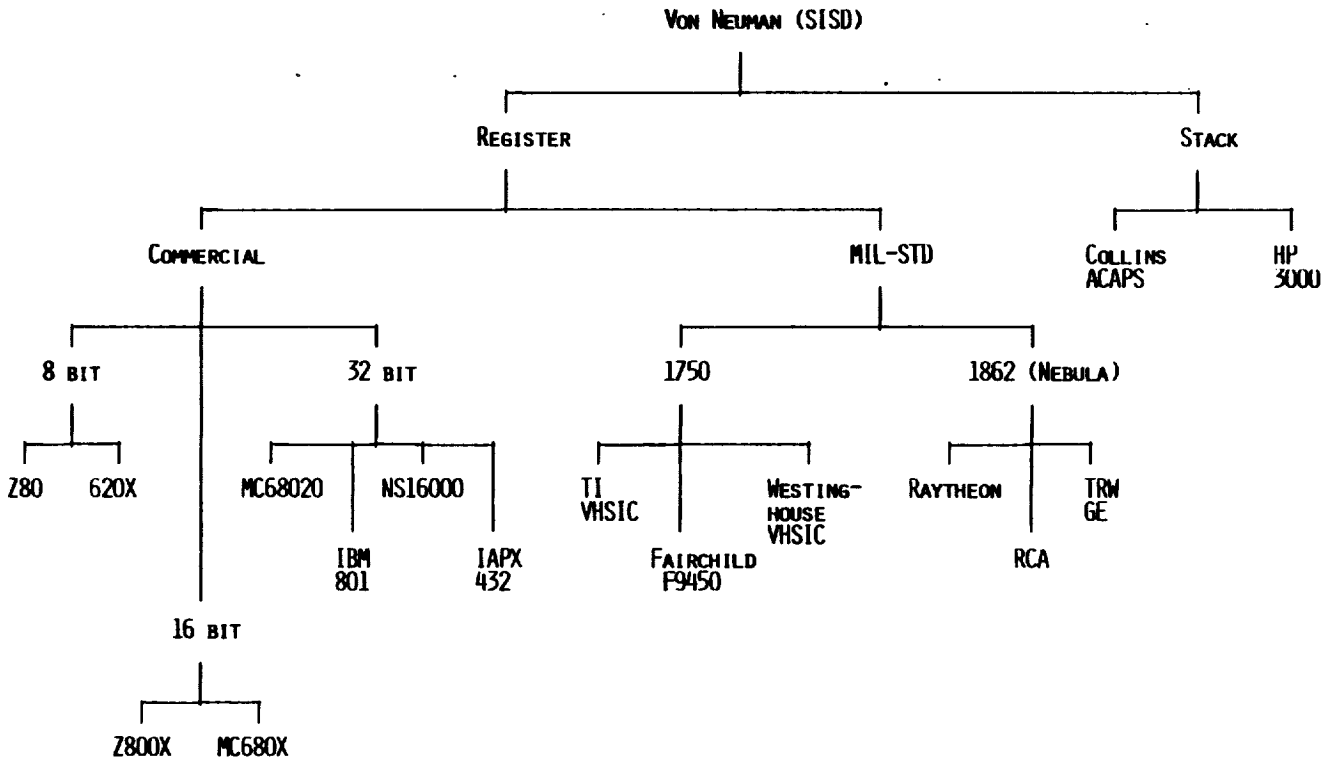
PROCESSOR ARCHITECTURE

- UNIPROCESSOR
  - PROTOTYPICAL SYSTEMS
    - DRAPER FTP
    - SIFT
    - CMU C.VMP
  - EXPERIMENTAL SYSTEMS
    - F-8 DFWS
    - A-7 DIGITAC
    - F-16 AFTI FCS
  - OPERATIONAL SYSTEMS
    - DRAPER/WG FTP
    - MD80 DFGS
    - B757/767 FCS
    - SHUTTLE DPS
    - BELL ESS No. 1A
    - F-18 FCS

PROCESSOR ARCHITECTURE (CONT.)

- MULTIPROCESSOR
  - EXPERIMENTAL SYSTEMS
    - FTMP
    - SIFT
    - BMD CROSSBAR
    - C.VMP
  - OPERATIONAL SYSTEMS
    - CM*
    - C.MMP
    - PLURIBUS


ISA ARCHITECTURE
VON NEUMAN (SISD)
NON-VON NEUMAN
SIMD
MIMD
TI ASC
ILLIAC IV
CRAY1
CDC 6600
IBM 360/91


VON NEUMAN (SISD)
REGISTER
STACK
COMMERCIAL
MIL-STD
COLLINS ACAPS
HP 3000
8 BIT
32 BIT
1750
1862 (NEBULA)
Z80
620X
MC68020
NS16000
IBM 801
IAPX 432
TI VHSIC
WESTING-HOUSE VHSIC
FAIRCHILD F9450
RAYTHEON
TRW
GE
RCA
16 BIT
Z800X
MC680X

PROCESSING ARCHITECTURE CONCLUSIONS

- While none of the systems surveyed satisfies system level AIPS requirements, elements of these will be considered for inclusion in AIPS.

- Several promising candidates at the processor level
  - Uniprocessor - Instruction Synchronized Processor
  - Multiprocessor - Instruction Synched Homogeneous Multiprocessor

- Several options available at the instruction level
  - MIL-STD 1750 - Fairchild F9450
  - Commercial 32-bit - MC 68020, NS 16000

NETWORKING ARCHITECTURE

- Topology
- Operational Alternatives
- ISO Open Systems Interconnect Model
- Software Interface

NETWORK TOPOLOGY

- Bus
  - Broadcast Bus (ARINC 429)
  - CMND/Response Bus (MIL-STD 1553B)
  - Contention Bus (Most LANs, FTMP)
- Ring
  - JSC Testbed
  - LAN (Prime Net, Corvus Omninet, Apollo Domain)
  - CSDL/FIAT IMC3
- Mesh
  - CSDL Mesh/F-8 DSPM
  - ARPANET
- Fully Interconnected
  - SIFT
- Tree
  - Wangnet

NETWORK OPERATIONAL ALTERNATIVES

- Packet Switched
  - ARPANET
  - SYTEK LOCALNET 20
- Circuit Switched
  - BMD Crossbar
  - CSDL Mesh
  - Bell Telephone

NETWORK OPERATIONAL ALTERNATIVES (CONT.)

- TIME DIVISION MULTIPLE ACCESS (TDMA)
  - MIL-STD 1553 (F18, F20, B1)
  - CSMA/CD (ETHERNET, WANGNET, Z-NET)
  - CSMA/CA (HYPERCHANNEL, OMNINET)
  - TOKEN PASSING (DOMAIN, OMNILINK, PRIMENET)
  - DISTRIBUTED POLLING (FTMP SYSTEM BUS)
- WAVELENGTH DIVISION MULTIPLE ACCESS (WDMA)
  - LARC FIBER OPTICS WDM
  - CABLE TV

ISO OPEN SYSTEMS INTERCONNECT MODEL

- APPLICATION
- PRESENTATION
- SESSION
- TRANSPORT
- NETWORK

} SOFTWARE INTENSIVE LAYERS

- DATA LINK
- PHYSICAL

} INTERFACE WITH PHYSICAL CHANNEL

NETWORK SOFTWARE INTERFACE

- Single Distributed Operating System

- Multiple Communicating Operating Systems

NETWORKING ARCHITECTURE CONCLUSIONS

- Topology

  Multiplex bus is vulnerable to faults and damage.

  Full cross-strapping is most secure.

- Operational Alternatives

  Token passing and polling are likely bus arbitration schemes.

  Circuit switching is preferable to packet switching.

COMPUTATIONAL CORE FDIR

- Temporal Checks
  - Watch Dog Timers (MD 80 DFGS, ESS No. 1A, SCCM)
  - Memory/Bus Timeouts (Most CPUs)
  - Instruction Retry (IBM 30xx, Most Mainframes)
- Integral Checks
  - Arithmetic Codes (JPL Star)
  - Data Path Parity (Mainframe CPUs)
  - Memory Parity (Most Minicomputers)
  - Hamming Codes (Mainframe Memories)
  - Morphic Logic (SCCM)
- Diagnostic Checks
  - ROM Checksum (767 FCS)
  - CPU Opcode Checks (Pluribus)
  - Voter Checks (FTMP)

COMPUTATIONAL CORE FDIR (Cont.)

- Replication, Comparison and/or Voting
  - Low Level Replication
    - Raytheon SFTC
  - High Level Replication
    - Approximate Replication
      - MD 80 DFGS, AFTI/F-16 FCS
    - Exact or Congruent Replication
      - FTMP, SIFT, Shuttle DPS, FTP, etc.
    - Stand-by Sparing
      - Bell ESS No. 1A, Voyager CCS

COMMUNICATION FDIR

- Temporal Redundancy
  - CRC Checks in LANS
  - Bose Chaudhri (BC) Codes in Communication Networks
- Spectral Redundancy
- Spatial Redundancy
  - Multiple Buses (Most Avionic Systems)
  - Mesh (Arpanet)

FAULT DETECTION, ISOLATION, AND RECOVERY CONCLUSIONS

- Replication is the most effective means of providing very high coverage.
- Integral checks are suitable for large memories.
- Temporal and diagnostic checks are helpful in uncovering latent faults.

ARCHITECTURE TECHNOLOGY SUMMARY

- Diverse Technology Base
- NASA, DOD, Industry, and Academia
- Operational and Prototypical Systems and Paper Designs

SOFTWARE TECHNOLOGY SURVEY

- High-Order Languages
- Fault-Tolerant Software
- Distributed Operating Systems
- Software Tools and Methods

HIGH-ORDER LANGUAGES

VISITS

MICHAEL RYER	AT	INTERMETRICS	(MAY 20)
JAMES LARUE	AT	SOFTECH	(JULY 26)
ED CHEVERS	AT	JSC	(AUGUST 9)
CHARLES MCKAY	AT	UHCL	(AUGUST 9)
MARK GERHARDT	AT	RAYTHEON - R.I.	(JULY 14)
RON COFFIN	AT	COLLINS	(JUNE 23)

TELECONS

TELESOFT

AJPO

ROLM

WESTINGHOUSE

VENDOR DEMO

COMPUTER THOUGHT

HIGH-ORDER LANGUAGES (CONT'D)

INTERMETRICS - ADA ACTIVITIES

- AIR FORCE
- IBM - VM
- APSE (AIE)
  - ON HOLD
  - SLIPPED ONE YEAR - 1984
- COMPILER
  - LATE 1984
- PDL - BYRON - AVAILABLE

SOFTECH - ADA ACTIVITIES

- ARMY
- VAX - VMS
- APSE (ALS)
  - DELIVERED 1982
  - DOCUMENTED
  - IN USE
- COMPILER
  - PRELIMINARY - OCTOBER 1983
  - FULL OPTIMIZED - JULY 1984

HIGH-ORDER LANGUAGES (CONT'D)

JSC/UHCL - ADA ACTIVITIES

- BETA TEST SITE FOR SEVERAL COMPILERS
- COURSES IN ADA

  GRADUATE LEVEL

  SENIOR UNDERGRADUATE

  RANK BEGINNERS

Alternatives to Ada-1

JOVIAL

- Compilers Available
- Change and Maintenance Supported by DOD
- Development and Test Tools Available

HAL

- Compilers Available
- Change Unsupported
- Some Tools Available

FORTRAN

- Compilers Available
- Change and Maintenance Supported
- Some Tools Available
- Lacks Modern Language Constructs
- No Realtime

Alternatives To Ada-2

C

- Compilers Available
- Change Unsupported
- Tools Available Within Unix

PASCAL

- Incomplete in some Respects
- Not Standard
- No Realtime
- Code Efficiency Issue

HIGH-ORDER LANGUAGES (CONT'D)

Summary

Use of Ada is Desirable

Production Ada Compilers not Here Now

AIPS Implementation Language Decision TBD

Softech APSE currently available

Intermetrics APSE not Currently Available

Ada PDL's are available

Target Machine Support is a Problem

HIGH-ORDER LANGUAGES (CONT'D)

What Should CSDL Do?

- Acquire Ada Compilers and Learn Ada
- Acquire and Use an Ada PDL
- Make Implementation Language Decision and Acquire Tools
- Acquire or Develop a Run-Time Environment

HIGH-ORDER LANGUAGES (CONT'D)

What Should NASA Do?

- Investigate Issues Surrounding Use of Ada in Distributed Processor Environment for Avionics Applications

FAULT-TOLERANT SOFTWARE

Visits

Herb and Myron Hecht at SOHAR (July 14)

Avizienis at UCLA (July 15)

McDonald at BMD (August 23)

Carter at IBM (September 15)

Telecons

Musa at Bell Labs

Basili at U. Maryland

Goel at Syracuse U.

FAULT-TOLERANT SOFTWARE (CONT'D)

Approaches Identified

Backup Software

Recovery Blocks

N-Version Programming

FAULT-TOLERANT SOFTWARE (CONT'D)

Resident Backup Software (Rebus)

- Under Development at CSDL for F-8
- Acceptance Test Failure Causes Switchover
- Emergency Capability Only
- Backup Mode in Effect Until Plane Lands
- Single Backup Version

FAULT-TOLERANT SOFTWARE (CONT'D)

Recovery Blocks and N-Version Programming

- Limited Applications to Date
- Multiple, Independent Programming Efforts Desirable
- Correct, Complete and Unambiguous Spec Desirable
- Absence of Correlated Failures
  (Taken on Faith/Not Formally Demonstrated)
- Cost Effectiveness Not Clear

FAULT-TOLERANT SOFTWARE (CONT'D)

Relevance to AIPS

- Backup Software Possibly Useful for AIPS
- Both Recovery Blocks and N-Version Programming have Potential to Significantly Improve SW Reliability
- Recovery Blocks Questionable in Real-Time Environment
  (However, not all AIPS Functions are Time Critical)
- N-Version Programming
  - Better Suited to Real-Time Environment
  - Ensures Functional Correctness

FAULT TOLERANT SOFTWARE (CONT'D)

What Should CSDL Do?

- Fault-Tolerant Software Experiment
- Software Reliability Data Gathering (Basili, McGarry)
- Hardware/Software Reliability Modeling (Goel, Musa)

FAULT-TOLERANT SOFTWARE (CONT'D)

What Should NASA Do?

- Cost Effectiveness of Fault Tolerant Software
  (Does Increased Reliability Justify Added Cost and Complexity?)
- Correlated Errors Among Multiple Program Versions
  (Both Recovery Blocks and N-Version Programming Assume Absense of Correlated Errors Among Independently Generated Versions. Is this Justified?)
- Applications Sensitivity of Fault-Tolerant Software
  (Which Applications are Most Suitable?)

DISTRIBUTED OPERATING SYSTEMS

Visits

Popek at UCLA (July 15)

Segall at Carnegie-Mellon (June 24)

Jensen at Carnegie-Mellon (June 24)

McDonald at BMD (August 23)

DISTRIBUTED OPERATING SYSTEM (CONT'D)

Examples

ARPANET	(BBN)
LOCUS	(UCLA and ICA)
NOS	(PLEXUS)
ARCHONS	(CMU)
ACCENT	(CMU)
GRAPEVINE	(Xerox)
Star OS	(CMU)
MEDUSA	(CMU)
ROSCOE	(Wisconsin)

DISTRIBUTED OPERATING SYSTEMS (CONT'D)

Potential Functions

- Task Dispatching/Scheduling
- Interprocess Communication
- Timing/Synchronization
- Fault Tolerance
- Restart Procedures
- Load Leveling
- Reconfiguration
- Data-Base Management
- Resource Management
- Backup/Archiving
- Multi-Level Memory Management

DISTRIBUTED OPERATING SYSTEMS (CONT'D)

Summary

- Distributed OS in Very Early Stages of Development
- SW Development and Testing "Horrendously Difficult"
- Distributed OS for a RT Environment does not Presently Exist
- May be able to Adapt Transaction-Oriented OS
- Conclusion: Distributed OS is Potentially High Risk

DISTRIBUTED OPERATING SYSTEMS (CONT'D)

What Should CSDL Do?

- Develop Framework for Design Choices
- Develop or Acquire Basic Distributed Operating System
- Add Additional Functions as Needed
- Address V&V Issues Early

DISTRIBUTED OPERATING SYSTEMS (CONT'D)

What Should NASA Do?

- Develop Taxonomy for Distributed Operating Systems and Evaluate for Avionics Applications

SOFTWARE TOOLS AND METHODS

TELECONS

ORGANIZATIONS	SUBJECT	KEY PEOPLE
AIRMICS	TARGET MACHINE SUPPORT FOR ADA	C. GIESE
SPERRY-UNIVAC	WALK THROUGH TECHNIQUES	J. HART
NASA-GSFC	S/W ENGINEERING DATA	F. MCGARRY
USAF-AFWAL	ENVIRONMENTS	I. CARO, R. SZYMANSKI, D. EHRENFRIED
NADC	PERFORMANCE SIMULATION	C. MATTES

SOFTWARE TOOLS AND METHODS (CONT'D)

VENDOR DEMOS

SUN MICROSYSTEMS, INC.

INTEGRATED SYSTEMS CORP.

HONEYWELL

VISITS

DATE	LOCATION	KEY PEOPLE
20 MAY, 24 AUGUST	INTERMETRICS, CAMBRIDGE, MA.	M. RYER
2 JUNE	NASA LaRC, HAMPTON, VA.	S. VOIGT
13 JULY	NASA JPL, PASADENA, CA.	T. RENFROW
14 JULY	NASA JPL, PASADENA, CA.	R. TAUSWORTHE
26 JULY	SOFTECH, WALTHAM, MA.	L. WEISSMAN

SOFTWARE TOOLS AND METHODS (CONT'D)

REQUIREMENTS PHASE

• NRL's Requirements Methodology	Used at NRL, CSDL
• Structured Analysis	Used at CSDL, Various Places
• PSL/PSA	Used at JPL; not Mature
• RSL/REVS	Used at JPL; not Mature
• DARTS	Used at CSDL; not Mature

SOFTWARE TOOLS AND METHODS (CONT'D)

DESIGN PHASE

• PDL	Caine, Farber and Gordon, Widely Used
• CRISP	Used at JPL, Available through COSMIC
• SDDL	Used at JPL, Available through COSMIC
• BYRON	Available from Intermetrics (IBM)
• DARTS	Available at CSDL (IBM)
• MASCOT	Availability Questionable, Uses Language CORAL
• USEIT	Available from HOS (VAX)
• GENSKEL	Softech Proprietary
• SADT	Available from Softech

SOFTWARE TOOLS AND METHODS (CONT'D)

Code, Unit Test, Validation Phases

• UNIX Tools	AT&T; Berkeley
• MASCOT	CORAL Specific, UK
• Structured Testing	McCabe Assoc.
• IVTS	HAL/S Specific, Under Development by LaRC
• LEKSNEG	Ada Specific, Softech Proprietary
• ALS	Operational
• AIE	Design Only

SOFTWARE TOOLS AND METHODS (CONT'D)

Management Tools

• Software Cost Estimation	
- COCOMO	Manual
- WICOMO	Automated (VAX)
- SOFTCOST	Automated (Modcomp)
• Configuration Control and Status Reporting	
- LIPSVC	CSDL
- MADNET	Modcomp
- UNIX Tools	E.G. MAKE, SCCS
- ALS	
- Manual Techniques	
• Software Quality Measurement	
- DARTS	Automated
- RADC's Measurement Techniques	Manual
- ARMY/RADC's Automated Measurement Tool	Automated, COBOL

SOFTWARE TOOLS AND METHODS (CONT'D)

What Should CSDL Do?

- General
  - Strive for "Integrated", not "Bundled" Tools which are based on a UNIX-Like Environment with Integrated Database

- Specific
  - Requirements: NRL's Methodology, Structured Analysis
  - Design: BYRON, DARTS, ALS, GENSKEL, UNIX Tools
  - Code/Test/Verification: LEKSNEG, Structured Testing, UNIX Tools, ALS
  - Management: WICOMO, DARTS Metrics, UNIX Tools (MAKE, SCCS), ALS

SOFTWARE TECHNOLOGY SURVEY

Conclusions

Key Items for CSDL Attention

- HOL Selection
- Distributed Operating System Design
- Target Machine Run-Time Environment
- Software Development Station Tools
- Ada PDL Acquisition
- Software Reliability Model Strategy
- Fault-Tolerant Software Experiment Definition

HARDWARE TECHNOLOGY SURVEY

- Processor Instruction Set Architecture
- Semiconductor Technology
- Data-Net Transmission Medium
- Power Distribution and Control
- Mass Memories

ORGANIZATIONS SURVEYED

ORGANIZATION	TOPIC	MODE
Fairchild	1750A, Semiconductor	Visit
Honeywell	1750A, Optical Couplers	Visit, Telecon
RCA	Semiconductor	Visit
IBM	1750A - AP101E	Literature
McDonnell Douglas	1750A	Literature
Mikros Systems	1750A	Literature
Tracor Aerospace	1750A	Literature
Raytheon	1862A	Visit
Intel	Semiconductor, Bubbles	Telecon, Literature
Western Electric	Bubble	Literature
Sandia	Semiconductor	Telecon

ORGANIZATIONS SURVEYED (CONT'D)

ORGANIZATION	TOPIC	MODE
TI	VHSIC	VISIT
VALTEC	FIBER OPTICS	TELECON
CAN STAR	OPTICAL COUPLERS	TELECON
FIBERLAN	FIBER OPTIC NETWORKS	TELECON
FREQUENCY CONTROL PRODUCTS	OPTICAL SWITCHES	TELECON
JPL	BUILDING BLOCK COMPUTER	VISIT
	POWER DISTRIBUTION	TELECON
JSC	MASS MEMORIES	VISIT
	POWER DISTRIBUTION, 1750A	TELECON
LARC	BUBBLES, OPTICS, VHSIC	VISIT

PROCESSOR INSTRUCTION SET ARCHITECTURE

- MIL-STD-1750A
- MIL-STD-1862A
- COMMERCIAL ARCHITECTURE

PROCESSOR INSTRUCTION SET ARCHITECTURE

- MIL STD 1750A
  - 16 Bit Architecture
  - Air Force Controlled and Maintained
  - Preproduction Stage
  - Implementations
    - Single Chip
    - Multi Chip
    - Single Board
    - Microprogrammed
  - Present Problems
    - Yield
    - Speed
    - Validation of Compliance to ISA
  - Limited Development Tools
  - Availability - Preproduction Late 1983
    Production Late 1984

PROCESSOR INSTRUCTION SET ARCHITECTURE (CONT'D)

- MIL STD 1862A
  - 32 Bit Architecture
  - Army Controlled and Maintained
  - Brassboard Stage
    - Validation of Compliance to ISA in Progress
  - Implementations
    - Single Board
    - Microcomputer
    - Minicomputer
  - Limited Development Tools
  - Availability 1986

PROCESSOR INSTRUCTION SET ARCHITECTURE (CONT'D)

- Commercial ISA
  - Continually Evolving
  - Manufacturer Controlled
  - Phaseout Potential
  - Development Tools Available
  - Implementations
    - Chips
    - Boards
    - Systems
  - Some Implementations Available on QPL

PROCESSOR INSTRUCTION SET ARCHITECTURE (CONT'D)

- Motorola
  - 68XXX Family
    - 16 Bit
    - 32 Bit
    - Coprocessors
      - Floating Point
    - I/O Chips
      - Periphial Controllers
      - Communication Protocol Controllers
      - Dual Port RAM

PROCESSOR INSTRUCTION SET ARCHITECTURE (CONT'D)

- Intel
  - 80XXX Family
    - 8 Bit
    - 16 Bit
    - Coprocessor
      - Floating Point
    - I/O Chips
      - DMA Controller
      - Bus Interface Controller
      - Communication Protocol Controller

PROCESSOR INSTRUCTION SET ARCHITECTURE (CONT'D)

- IAPX 432
  - 32 Bit
  - Compiler Oriented
  - Internal Floating Point
  - Supports Concurrent Programming and Execution
  - Built in Hardware Run-Time Protection
  - Self Checking Mode
  - Transparent Multiprocessing

PROCESSOR INSTRUCTION SET ARCHITECTURE

- Conclusions
  - MIL-STD-1750A Processors Will Probably be Available Early 1984
  - MIL-STD-1862A Processors will not be Available Until 1986
  - Commercial ISA Processors are Available now but are Subject to Manufacturer Discontinuance

SEMICONDUCTOR TECHNOLOGY

SEMICONDUCTOR TECHNOLOGY

PRESENT TECHNOLOGY

TECHNOLOGY	GATE DELAY (TYP)	POWER/GATE (TYP)	RAD TOLERANCE (TOTAL DOSE)	
TTL	10 NS	10 MW	10^6	
SCHOTTKY TTL	5 NS	20 MW	10^6	
I^2L	10 NS	60 NW	10^6	
SCHOTTKY I^2L	2 NS	30 PW	10^6	
ECL	1 NS	20 MW	$>10^7$	
GA AS	50 PS	5 MW	10^8	BUFFERED FET
PMOS	120 NS	1 MW	?	
NMOS	50 NS	0.2 MW	10^3-10^4	
HMOS	1 NS	1 MW	10^3-10^4	
CMOS	50 NS	10^{-4}-10 MW	10^5-10^6	4000 SERIES
VHSIC	2 NS	30UW	10^6	INTEGRATED SCHOTTKY LOGIC

SEMICONDUCTOR TECHNOLOGY (CONT'D)

DEVELOPING TECHNOLOGY

- CMOS
  - LINE WIDTHS OF ORIGINAL CMOS PERMITTED FEWER AND SLOWER DEVICES PER DIE
  - NEW LINE WIDTHS
    - DENSITY APPROACHING NMOS
    - LOWER POWER DISSIPATION THAN NMOS
    - PROPAGATION DELAYS < 8 NS
    - WIDE TEMPERATURE RANGE
    - LESS STRINGENT POWER SUPPLY REQUIREMENTS
    - HIGH NOISE IMMUNITY
    - AVAILABILITY
      - DIRECT REPLACEMENT TTL
      - DIRECT REPLACEMENT MICROPROCESSORS

SEMICONDUCTOR TECHNOLOGY (CONT'D)

- VHSIC
  - In Early Development Stages
  - Limited Products Initially Developed for DOD Projects
    - Arithmetic Units
    - Multiplier
    - Memories
    - Gate Arrays
  - High Speed

SEMICONDUCTOR TECHNOLOGY

- Conclusions
  - The Semiconductor Industry is Converting Much of its NMOS Technology to CMOS
  - VHSIC would be Useful if it Were Available

DATA - NETWORK TRANSMISSION MEDIUM

- Copper Wire
- Fiber Optics

DATA - NETWORK TRANSMISSION MEDIUM

- Copper Wire
  - Mature Technology
  - Transmission Characteristics Known
  - Environmental Requirements Easily Met
  - Interconnection Technique Simple
  - Wire Adds Considerable Weight
  - Generates and Susceptible to EMI

DATA - NETWORK TRANSMISSION MEDIUM (CONT'D)

- Fiber Optics
  - Less Mature Technology
  - Electrical Isolation Between Systems
  - Immunity to Electromagnetic Interference
  - Light Weight
  - High Bandwidth

DATA - NETWORK TRANSMISSION MEDIUM (CONT'D)

- Wave Length Division Multiplexing Being Investigated by LaRC
  - Multichannel
    - Simultaneous Multiple Users
    - Simultaneous Multiple Formats
    - Dedicated Channels
      - Clock
      - Broadcast Bus
  - Multidirection
  - Passive Mux - Demux Possible
  - Increased Bandwidth

DATA - NETWORK TRANSMISSION MEDIUM (CONT'D)

- Environmental Concerns
  - Radiation
    - Increasing Transmission Losses
  - Vibration and Shock
    - Effects on Interconnections
    - Effects on Alignment
  - Temperature
    - Effects of Expansion and Contraction
    - Operation at Temperature Extremes

DATA - NETWORK TRANSMISSION MEDIUM (CONT'D)

- Laser Diodes
  - Monochromatic
  - Low Spectral Width for less Loss and High Bandwidth
- LEDS
  - Less Monochromatic
  - Lower Bandwidth capability than Laser Diodes
  - High Spectral Width
- Studies in Fiber Optic System at LaRC
  - Optical Node
  - Optical Switch

DATA - NETWORK TRANSMISSION MEDIUM (CONT'D)

- Couplers
  - Passive Star
    - Loss Proportional to Number of Ports
  - Tee
    - Loss Cumulative
  - Switches
    - Electromechanical

DATA - NETWORK TRANSMISSION MEDIUM

- Conclusions
  - Coppper Wire is a Mature Technology
  - Fiber Optics is Maturing but still has Some Problems

POWER CONTROL AND DISTRIBUTION

- Considerable Work has been done on Power Generation and Conversion
- Little Published on Control and Distribution for Fault Tolerance, Redundancy, Etc.
- Examples of Power Distribution Systems
  - Local Power Grid
  - Telephone Companies
  - FTMP
    - Quad Primary Power Sources
    - Load Sharing From Primary Sources at LRU Power Supply
  - Sift
    - Quad Primary Power Sources
    - Diode "or" of Power to LRU
  - Aircraft
    - Independant Power Buses
    - Critical Power Bus
    - Power Bus Switching After Fault

POWER CONTROL AND DISTRIBUTION

- Topics to be Studied Further
  - Connection Techniques
    - Multiple Power Buses
    - Power Networks
    - Load Sharing
    - Diode Coupling
  - Fault Detection Algorithms
  - Fault Recovery Techniques
  - Automated Power Systems
    - Minimize Response Time
    - Autonomous Operation
      - Detection, Isolation and Recovery Performed Locally

POWER CONTROL AND DISTRIBUTION

- Conclusions
  - More Work has to be Done in Investigating Automated Control

MASS MEMORIES

- MAGNETIC TAPE
- MAGNETIC BUBBLES
- DISC
  - MAGNETIC
  - OPTICAL
- SEMICONDUCTOR

MASS MEMORIES

- TAPE
  - MATURE TECHNOLOGY
  - LARGE CAPACITY
  - LONG ACCESS TIME
  - SUBJECT TO MEDIUM CAUSED ERRORS
  - CONTAMINATION OF HEADS
  - MECHANICAL WEAR
  - LIMITED TAPE USE CYCLES
- BUBBLES
  - EVOLVING TECHNOLOGY
  - NONVOLATILE
  - LIGHT WEIGHT
  - READ/WRITE CYCLES DO NOT CAUSE WEAROUT
  - NO POWER DRAIN IN STANDBY MODE
  - RADIATION TOLERANCE DEPENDENT ON CONTROL CIRCUITRY
  - DENSITY INCREASING

MASS MEMORIES (CONT'D)

- Disc
  - Magnetic Disc
    - Problems of Operation in Severe Environment
    - Mechanical Wear
  - Optical Disc
    - Density Greater than Magnetic Disc
    - Same Problems as Magnetic Disc
- RAM
  - Highest Speed
  - Density Continues to Evolve
  - Volatile
  - CMOS RAMS
    - Increasing Speed and Density
    - Battery Backup

MASS MEMORIES

- Conclusions
  - Tape is Available for High Storage Requirements but has Wear Problems
  - Bubbles are Improving in Density
  - Discs for Severe Environments Must be Looked At
  - RAMS are Increasing in Density

METHODOLOGIES

SURVEY

SCOPE

- Hardware Reliability, Maintainability, Availability (RMA)
- Software Reliability
- Computer System Performance
- Performability - A Combined Consideration of Performance and Reliability

GOALS

- Comparison of Generic Approaches
- Definition of Measures of Merit
- Comparison of Evaluation Mechanizations

RMA SUMMARY

SURVEY SOURCES

Source	Subject	Contact Mode
NASA Langley/S. Bavuso,	Care III	Visit
Duke Univ./K. Trivedi	Harp (Hybrid Automated Reliability Predictor) Care III	Visit
Carnegie-Mellon/Z. Segall	Relicalc, Stars Transient Faults	Visit
Lockheed-Georgia/ W. Ness	Carsra, Care II, Aries, Cast, Tasra, Care III	Telephone
Sequoia Systems/J. Stiffler	Care III	Telephone
Batelle Lab/M. Bridgeman	Tasra, Fault Tree	Telephone
Boeing Military/A. Stern	Fault Tree Hybrid	Telephone
CSDL	Mark 1	Visit
IEEE Trans	Aries, Networks	Literature
Symposium on Fault Tolerant Computing	Surf	Literature

STATUS

- Several Approaches Have Been Thoroughly Studied by Different Organizations
- Several Computer Mechanizations are Available
- The Application of the Mechanizations to Practical Problems has been Demonstrated
- Some Comparison Studies among the Mechanizations have been done

ISSUES

- Automated Versus Hybrid Mechanizations
- Markov Models Versus Fault Tree Analyses
- Failure and FDI Error Models
- Evaluation of Communication Networks

CONCLUSIONS

- The Preferred Measure of Merit is Failures/Million hours given the Maintenance Interval
- The preferred Evaluation Approach is the Markov Model
  - Used in Almost all Computer Mechanizations
  - More versatile than Fault Tree Models
  - Only Feasible Approach to Evaluating Availability
  - Techniques Exist for Bounding the Number of States
- The Evaluation of Network Interconnection Reliability has Priority
  - The Computer Interconnections will Probably Dominate the Overall Reliability
  - Minimal Practical Work has been done in this area

CONCLUSIONS (CONT'D)

- In-House Hybrid Approach (Mark 1) will be used for the Architecture Evaluation
  - Verified by NASA LaRC and LAAS (Surf)
  - Broad Experience with the Mechanizations and its use
  - Adaptable to new Designs
  - Minimum need for External Help
- Areas of Outside Support
  - Now Constant Failure Rates
  - Failure Models (Latent Failures)
  - An Independent RMA Assessment of the AIPS Architecture

SOFTWARE RELIABILITY
SUMMARY

SURVEY SOURCES

Source	Subject	Contact Mode
Bell Labs/J. Musa	Reliability Models	Telephone
CSDL	Markov Models Data Gathering	Visit
Syracuse U./A. Goel	Reliability Models	Telephone
Maryland/V. Basili	Data Gathering	Telephone

STATUS

- Several Models have been Suggested
- Minimal Data Exists to Substantiate the Models
- Widely accepted Computer Mechanizations do not Exist
- Application to Realistic Programs is Limited
- Few Comparison Studies Among the Models Exist

CONCLUSIONS

- This Technology is not Mature Enough for Application to AIPS on an Absolute Basis
- Area of Outside Support
  - Data Collection to Support Models
  - Develop a Software Reliability Evaluation Methodology for Real-Time, Redundant, Distributed Systems

PERFORMANCE SUMMARY

SURVEY SOURCES

Source	Subject	Contact Mode
Duke Univ/K. Trivedi	Queuing Theory	Visit
Carnegie-Mellon/Z. Segall	Simulation	Visit
NADC/C. Mattes	Generalized Computer System Simulator (GCSS) II	Telephone
BMD	Test Bed	Visit
D. Ferrari	Analytic Approaches	Book
B. Beizer	Semi-Deterministic	Book
D. Siewiorek	Analytic Approaches	Book
U.S. Army Electronics Command	Computer Family Architecture Selection	Report
National Comp. Conf	All Methods	Literature
IEEE Trans	All Methods	Literature
Symp. on Fault Tolerant Comp	All Methods	Literature

STATUS

- Several Approaches have been Developed
  - Semi-Deterministic
  - Stochastic
  - Simulation
- Several Practical Problems have been Solved
  - Primarily Transaction Type Systems Rather Than Real-Time Systems
- Very few Computer Mechanizations are Available
- Comparison between Approaches is Heuristic Only

CONCLUSIONS

- Preferred Measures of Merit are Throughput, Delay Time and % Overhead
- The Analytic Approaches are the Semi-Deterministic Approach and Queuing Theory
  - The Semi-Deterministic Approach
    - Will be Used Initially on AIPS
    - Some the the AIPS Architectures have Nearly Deterministic Workloads (Dedicated Hierarchical)
    - Based on Delay Models as a Function of throughput
    - Randomness will be Dealt with Using Mean Values
  - Queuing Theory
    - Stochastic Approach
    - Requires Characterization of Workload Statistics
    - Not well Suited to Shared Resources

CONCLUSIONS (CONT'D)

- GCSSII and BMD Test Bed will be Evaluated
- Area of Outside Support
  - Develop a Performance Evaluation Methodology for Real-Time, Redundant, Distributed System

PERFORMABILITY SUMMARY

SURVEY SOURCES

Source	Subject	Contact Mode
Univ. of Mich/J. Meyer	Performability Theory and Applications	Visit
CSDL	IUS	Visit
Batelle Lab/M. Bridgeman	Comparison of Performability and Reliability Evaluation Approaches	Report

STATUS

- Only Basic Ideas have been Defined
- Very Limited Practical Experience Exists
- Only a Single Mechanization Exists

CONCLUSIONS

- Performability is the Proper Measure of Merit for Fault Tolerant Systems
- It is not Necessary to use Performability During the Early Stages of System Design
- An effort will be made to Combine the Reliability and Performance Measures for the next Phase of AIPS
- Area of Outside Support
  - Develop a Performability Application Methodology for the Spectrum of NASA Applications

V&V SURVEY

VERIFICATION

THE PROCESS OF ESTABLISHING THAT THE DEVELOPED ARTICLE (H/W, S/W, SYSTEM) SATISFIES BOTH THE SYSTEM REQUIREMENTS AND ITS INDIVIDUAL REQUIREMENTS AT EACH STEP OF THE DEVELOPMENT.

C.4

VALIDATION

THE PROCESS OF ESTABLISHING THAT THE DEVELOPED PRODUCT SATISFIES THE SYSTEM REQUIREMENTS AND COMPLIES WITH THE VEHICLE REQUIREMENTS. THE OBJECTIVES ARE TO DEMONSTRATE THIS COMPLIANCE UNDER OPERATING CONDITIONS AND THE ABSENCE OF UNDESIRED EFFECTS. THE PROCESS MAY INCLUDE SIMULATION AND/OR FLIGHT TEST.

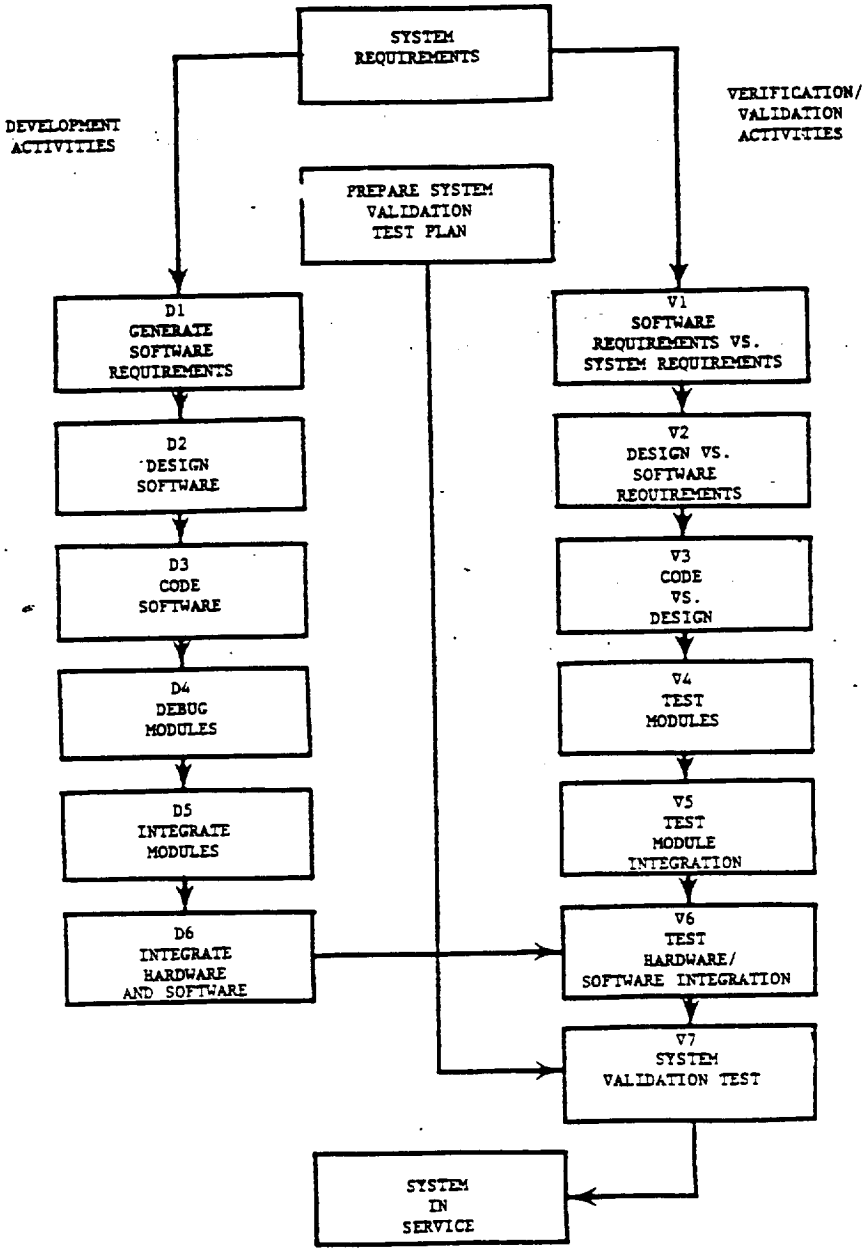
CONTACTS	SUBJECT	REMARKS
DUKE UNIV./K. TRIVEDI	HARP/CARE III	
CARNEGIE-MELLON/D. JENSEN	DIST. OP. SYS.	• HAS NOT CONSIDERED V&V
JPL/R. LOESH T. BRADY	GALILEO V&V	• HORRENDOUS PROBLEM • SCHEDULE PROBLEMS • DOLLAR PROBLEMS • NO HOOKS • NOT INCREMENTAL
SOHAR/H. HECHT M. HECHT	FAULT TOLERANT S/W HARDWARE AIDS	• "HOOKS" • STACK REGISTERS

Contact	Subject	Remarks
UCLA/A. Avizienis	Fault Tolerant S/W	• N-Version Self Validating • Would like to Compare BFS to PFS
ICA/G. Popek	Dist. Op. Sys.	• Horrendous to Test
BCAC/D. Prongay K. Heidergott I. Reese D. Prada	757/767 V&V Certification	• LRU V&V by Vendor • Requirements Problems • Schedule Problems • FMC V&V Fragmented • Blue Label
BMD/W. McDonald	BMD Testbed	• V&V on Op. Sys. by Use • V&V on BMD System Automated Fault Injection and Statistical Analysis

Contact	Subject	Remarks
Intermetrics (Seattle) J. Hanaway D. Brown	757/767 V&V	• Contractor to BCAC • Requirements Problems • Blue Label
CSDL/Various	Apollo	• Concept - Flight • H/W - S/W - System
	Shuttle	• PFS CFT • BFS Design/Test
	DMSP	• IVV
	FTP	• Design - Demonstration
	F-8	• Design - Flight
	Trident	• Design - Flight

SCOPE

- SYSTEM
- HARDWARE
- SOFTWARE



SOFTWARE VERIFICATION AND VALIDATION ACTIVITIES

SYSTEM V&V

- SHUTTLE - SAIL, SMS, SVDS, FSL
- 757/767 - CS AND FCL
- BMD - TESTBED
- F-8 - IRON BIRD
- FTMP - SIM, AIRLAB

HARDWARE

- FTMP - FAULT INJECTOR, FMEA
- F-8 - FMEA/FMET

SOFTWARE

- SHUTTLE - SDL/SPF, SLS, FSL
- GALILEO - INTEGRATED TEST
- DMSP - MSTF/DSS
- TRIDENT - SIMULATOR

ISSUES

SYSTEM	• RELIABILITY DEMONSTRATION • REQUIREMENTS • STRESS TESTING • MODELLING
HARDWARE	• FMEA/FMET • FAULT INJECTION • MODELING
SOFTWARE	• AUTOMATED TESTING • SIMULATOR/EMULATOR • TOOLS/TECHNIQUES • IV&V
ALL	• BUILT IN HOOKS • AIDS FOR TESTING, DOCUMENTATION, DATA COLLECTION • CLOCK CONTROL

LEARNED

- There are Many Development "tools" (Automated Procedures) Available Depending on Language
- There are Many Techniques for V&V of Fault Tolerant Systems
- There are no Portable, Directly Applicable "tools" for AIPS V&V
- There are no Tools or Techniques for V&V of Distributed or Decentralized Operating Systems
- V&V Efforts Suffer from:
  - Requirements Deficiencies
  - Schedule Inequities
  - Insufficient Resources
- Best Guide Today RTCA/DO-178

TECHNICAL IMPROVEMENTS

- Automation of Document/Report Generation
- Provide "Hooks" in Both S/W and H/W

TO BE DEVELOPED

- Automation for (Requirements → Specification → Code) Chain
- Realizable "Tool" for Validating Decentralized Operating System

SUMMARY

- Are Aware of the Majority of Available and Developing Technologies Appropriate to AIPS
- Have Identified Candidate Technology Developments to NASA (Meeting at NASA/Hqtrs August 26)
- Will Continue to Track Relevant Technologies
- The AIPS Design will Proceed Using Available Technologies (or Technologies to be Available in the Very Near Term)
- Have Ruled out Certain Technologies as being
  1) Too Immature
  2) Unavailable
  3) Undevelopable in the Required Time Frame
- Technology Decisions will be Made when Required for Design Decisions

LIST OF REFERENCES

The references listed below are those specifically called out in the text of this report. These references may be also among those entered in the Literature List, Appendix B.

(1) Philip H. Enslow, Jr., "Distributed Data Processing --- What Is It?", AGARD Conference Proceedings No. 303, October 1981.

(2) M. J. Flynn, "Very High Speed Computing Systems", Proc. IEEE, 54, December 1966, pp 1901-1909.

(3) L. Gmeiner and U. Voges, "Software Diversity in Reactor Protection Systems: An Experiment", IFAC Workshop SAFECOMP 1979, Stuttgart, 1979.

(4) R. Taylor, "Redundant Programming In Europe", ACM SIGSOFT SEN, Vol. 6, No. 1, January 1981.

(5) W. Ness, W. McCrary, and M. Bridgeman, "Automated Reliability and Failure Effects Methods for Digital Flight Control and Avionics Systems - Volume 1: Evaluation", NASA CR-166148, Lockheed-Georgia Co. - Battelle Columbus Lab, March 1981.

(6) E. G. Gai and P. R. Motyka, "Report of Visit with Sal Bavuso of NASA Langley Research Center", AIPS Internal Memorandum to Bob O'Donnell, AIPS-83-33, June 9, 1983.

(7) J. G. McGough, F. Swern, and S. J. Bavuso, "Methodology for Measurement of Fault Latency In a Digital Avionic Miniprocessor", AGARD Conference Proceedings No. 303, October 1981.

(8) P. R. Motyka, E. G. Gai, and R. E. Werner, "Technology Survey - K. Trivedi and R. Geist, Duke University", AIPS Internal Memorandum to R. O'Donnell, AIPS-83-39, Rev. 1, July 5, 1983.

(9) K. Trivedi and J. Clary, "Validation Methods Research for Fault-Tolerant Avionics and Control System Sub-Working Group Meeting - CARE III Peer Review", NASA Conference Publication 2167, Research Triangle Park, North Carolina, September 1980.

(10) S. R. McConnel and Daniel P. Siewiorek, "The Measurement and Analysis of Transient Errors In Digital Computer Systems", 9th Annual International Symposium on Fault-Tolerant Computing, June 1979.

(11) J. B. DeWolf, R. E. Werner and P. R. Motyka, AIPS Internal Memorandum to R. O'Donnell, AIPS-83-51, August 2nd, 1983. (Report on Visit to Carnegie Mellon and Meetings With Z. Segall, M. Satyanarayanan and D. Jensen).

(12) E. F. Hitt, M. Bridgeman and A. Robinson, "Comparative Analysis of Techniques for Evaluating the Effectiveness of Aircraft Computing Systems", NASA CR-159358, Battelle Columbus Laboratories, April 1981.

(13) A. Stern, "Stage - State Reliability Analysis Technique", AGARD Conference Proceedings No. 303, October 1981.

(14) Y. Ng and A. Avizienis, "A Unified Reliability Model for Fault-Tolerant Computers", IEEE Transactions on Computers, Vol. C-29, No. 11, November 1980.

(15) A. Costes, J. Doucet, C. Landrault and J. Laprie, "SURF - A Program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems", 11th Annual International Symposium on Fault-Tolerant Computing, June 1981.

(16) J. Barton DeWolf, F. C. Furtek, and E. G. Gai, "Reliable Software for Fault-Tolerant Systems", CSDL Report C-5487, April 9th, 1982.

(17) J. D. Musa, "Software Reliability Measurement - The State of The Art", Proceedings of Eurocon '82, 14-18 June 1982, Copenhagen, Denmark, North-Holland Publishing Company, Amsterdam, The Netherlands.

(18) D. Ferrari, "Computer System Performance Analysis", Prentice-Hall, 1978.

(19) Daniel P. Siewiorek and Robert S. Swarz, "The Theory and Practice of Reliable System Design", Digital Press (Digital Equipment Corporation), 1982.

(20) B. Beizer, Microanalysis of Computer System Performance, Van Nostrand Reinhold, Wokingham, England, 1978.

(21) J. V. A. Harrison, K. C. Daly, and E. G. Gai, "Reliability and Accuracy Prediction for a Redundant Strapdown Navigator", AIAA Journal of Guidance and Control, Vol. 4, No. 5, September-October 1981.

(22) J. Meyer, D. Furchtgott, and L. Wu, "Performability Evaluation of The SIFT Computer", IEEE Transactions on Computers, Vol. C-29, No. 6, June 1980.